

Master's Thesis

Juho Luoma

**Finite mixture models in comparison to k-means clustering in
both simulated and real world data**

Tampere University
Faculty of Information Technology and Communication Sciences
Master's thesis
Supervisors: Tapio Nummi, Jaakko Peltonen
May 2019

LUOMA, JUHO: Finite mixture models in comparison to k-means clustering in both simulated and real world data

M.Sc. Thesis, 51 pages

Tampere University

Master's programme in Computational Big Data Analytics

May 2019

Abstract

Finite mixture models are finite-dimensional generalizations of probabilistic models, which express the existence of groups or sub-populations that form the sample. In this thesis, multivariate normal mixtures are examined and compared to k-means clustering in different experimental situations. The comparison is carried out by simulations and by using a real-world, repeated measurements data set. A special extension of k-means clustering, *k-means for longitudinal clustering* (KmL), is used for the longitudinal data set. The goal of these experiments is to investigate if there is evidence to suggest that one method is better in some respect than the other.

Simulations were conducted to test the performance of the methods when increasing the number of outliers, average overlap between the clusters, the number of dimensions, and the number of observations. The data used in this thesis were collected as a part of iLiNS project which studied the effects of nutrient supplement to children's growth and mothers' health in rural areas of Malawi. There were 1391 Malawian mothers enrolled to the study, and the data consist of their children who were measured seven times from birth up to 30 months after birth.

In simulations, while requiring a non-random initialization for the algorithm, mixture models performed better than or equally well as k-means clustering in terms of correctly clustered individual data points. The parameter estimates by mixture models were also closer than or equally close to the true cluster centers as estimates by k-means. In real data, the participants were divided into clusters based on weight, using all the time points except the last one to form the clusters. The last measurement point was used to determine the status of growth for the child at 30 months. The dependency between the cluster identity of a participant and the growth status at last time point was tested with the χ^2 test. Both approaches were able to yield clusters that were formed so that the cluster membership of a participant was significantly related to growth status at 30 months, although the optimal number of clusters differed between the methods.

Key words Multivariate normal mixtures, KmL, R, mclust, longitudinal data

The originality of this thesis has been checked using the Turnitin OriginalityCheck service.

Contents

1	Introduction	7
2	Methods	9
2.1	Non-parametric clustering	9
2.1.1	K-means clustering	9
2.2	Finite mixture models	9
2.2.1	Maximum likelihood fitting and EM algorithm	13
2.2.2	Multivariate normal mixtures	15
2.2.3	Model selection	17
2.2.4	Practical example of the theoretical differences between the methods	19
2.3	Clustering of longitudinal data	21
2.3.1	Mixture modeling in longitudinal data	21
2.3.2	KmL: k-means clustering for longitudinal data	21
2.4	Earlier comparisons between model-based clustering and k-means clustering	22
3	Experiments on simulated data	24
3.1	Simulating the data	24
3.1.1	Generating the data	27
3.2	Performance on simulated data	27
3.2.1	Performance in terms of cluster purity	28
3.2.2	Performance in terms of distance between a parameter esti- mate and the true cluster center	34
4	Analyzing real data with clustering methods	43
4.1	Motivation	43
4.2	Data	43
4.2.1	iLiNS-DYAD	43
4.2.2	Preprocessing the data	43
4.3	Results	44
4.3.1	Mixture modeling	44
4.3.2	KmL	49
5	Conclusion	52
	References	54
	Appendix A: Histograms for normality	57

1 Introduction

Clustering is a widely used method for exploratory data analysis. It provides a way to examine data and it can be used for reducing dimensionality of the data, finding classes for data by common traits of the observations, and for creating research hypotheses. Obtained clusters can also be used as variables in predictive models.

There is more than one way to divide observations into separate groups and these methods can be divided into parametric methods and non-parametric methods. A parametric approach in question in this thesis, mixture model clustering, assumes that the data comes from K different sources or sub-populations each having their own distribution. Hence, the assumed distribution of the entire population is a mixture of distributions of the sub-populations. The task with mixture models is to identify the assumed sub-populations from which the data can be said to be drawn. A non-parametric approach makes no assumptions of the latent model behind the observations and forms the clusters based solely on the data. One such method is k-means clustering, and in this thesis, it is chosen as a point of reference to model-based clustering.

There has been some comparison between parametric and non-parametric approaches, and Gaussian Mixture Models are able to outperform k-means in clustering in certain situations (eg. VanderPlas 2016, chapter 5). However, it is possible that finite mixture modeling methods face difficulties with some anomalies in data that k-means could handle, and vice versa. To examine the performance of the chosen parametric and non-parametric approach, different simulations are carried out to test what kind of circumstances lead to better clustering with multivariate normal mixture models and k-means clustering.

Also, an implementation of k-means clustering designed specifically for longitudinal data has been developed and it has been shown to perform well when compared to a SAS (Statistical Analysis Software) procedure by Jones et al. (2001) of group-based trajectory model (Genolini & Falissard 2010). However, further comparison between k-means clustering for longitudinal data (*KmL*) and finite mixture models is reasonable, as there is more to finite mixture modeling in longitudinal data than just a group-based trajectory model. As the group-based trajectory model that Genolini & Falissard (2010) compared to *KmL* is only one case of multivariate mixture models, a more comprehensive utilization of multivariate mixture models could yield results that differ from the comparison between trajectory analysis and *KmL*.

The assumptions that this thesis seeks to test via simulations is under what circumstances, if at all, mixture models can possibly outperform k-means or vice versa. The thesis also attempts to extend the comparison between k-means and mixture models to longitudinal data in a way that uses mixture models in other purposes than trajectory analysis. Doing this in actual data, clustering solutions can be provided for the data at hand in more than one way.

The longitudinal context for testing the methods is achieved by using real world data. The data set that is used in this thesis was collected as a part of the iLiNS project

which studied the effects of nutrient supplement to mothers' and their children's health (Ashorn et al. 2014). The growth of children, closely related to health and well-being, is an important issue and it is possible to utilize clustering results for better understanding the growth development of a child and issues related to it. Furthermore, having insight on the development of child's growth provides a possibility to map out possible inhibiting factors of child's growth. The data is introduced further in chapter 4.

The aim of this thesis is to investigate mixture models, more precisely multivariate mixture models, and for reference compare them to k-means clustering. Methods are compared by simulating data and by applying methods to a real world repeated measurements data set. The aim of these simulations is to focus primarily on the correctness of the assigned data points in situations where the number of clusters is known. With the real world data, the focus is on finding the optimal number of components with each method and studying how well the clusters can be used in simple prediction task.

The theoretical background of the methods will be addressed in chapter 2. Chapter 3 introduces data simulation and the comparison on simulated data is discussed as well. In chapter 4, the aforementioned non-simulated data from Malawian children is examined in more detail and its characteristics are presented. Conclusions and final thoughts are presented in the last chapter.

The analyses, data preprocessing and part of simulations in this thesis are conducted with R software version 3.4.4. Part of simulations are conducted with the Python language (Python 3). The R package used for analyzing the real world data is *mclust*.

2 Methods

2.1 Non-parametric clustering

2.1.1 K-means clustering

The k-means algorithm is a very prominent clustering method, and although the idea behind the algorithm was presented as early as in the late 1950's, the name k-means was originally used by James MacQueen in 1967 (Bock 2008; MacQueen 1967). In order to perform k-means clustering, one has to decide the number of clusters beforehand to give as an input for the algorithm. For output the k-means algorithm yields a set of clusters. The standard k-means clustering algorithm has the following steps:

1. Select k objects randomly as initial centroids.
2. Compute the distance between each object x_i and the centroid m_j of each cluster j , and assign every object to its nearest cluster center. Distance measure commonly used is Euclidean distance and it is also used here. It is given by:

$$(2.1) \quad d(x_i, m_j) = \sqrt{\sum_{s=1}^n (x_{is} - m_{js})^2}, \quad s = 1, \dots, n.$$

where $d(x_i, m_j)$ is the distance between data point x_i and centroid m_j and n is the number of dimensions.

3. Calculate the mean of data assigned to each cluster to create the updated cluster centers
4. Repeat steps 2 and 3 until convergence.

(MacKay 2003, Chapter 20)

The idea behind forming the clusters with k-means has remained the same but the algorithm developed by Hartigan and Wong (1979) was an improvement to the original algorithm and is used as the R program's default algorithm for k-means clustering.

2.2 Finite mixture models

Finite mixture modeling is a statistical method for representing sub-populations within a population. That means, that the data at hand is not assumed to be from a single distribution but rather from a mixture of distributions (McLachlan & Peel 2000).

To consider finite mixture models, one denotes Y_1, \dots, Y_n a random sample of size n , Y_j being a random vector with p -dimensions that has probability density

function $f(\mathbf{y}_j)$ on the real coordinate space of p dimensions \mathbb{R}^p . Practically, \mathbf{Y}_j has observations of the random variables that correspond to p measurements made on the j th recording of some characteristics on the phenomenon that is being studied. Here, the realization of a random vector is denoted by lower-case letter in accordance with McLachlan & Peel (2000). For example, $\mathbf{y} = (\mathbf{y}_1^T, \dots, \mathbf{y}_n^T)^T$ denotes the observed random sample from the entire sample $\mathbf{Y} = (\mathbf{Y}_1^T, \dots, \mathbf{Y}_n^T)^T$. Superscript T denotes vector transpose. The basic form of the model is

$$(2.2) \quad f(\mathbf{y}_j) = \sum_{i=1}^k \pi_i f_i(\mathbf{y}_j).$$

(McLachlan & Peel 2000)

In equation (2.2), the π_i are the mixing proportions or weights and the $f_i(\mathbf{y}_j)$ are the densities. The mixing proportions are non-negative numbers that sum to one. This means that

$$(2.3) \quad 0 \leq \pi_i \leq 1 \quad (i = 1, \dots, k)$$

and

$$(2.4) \quad \sum_{i=1}^k \pi_i = 1.$$

Despite assuming feature vector \mathbf{Y}_j to be a continuous valued random vector, we can take $f(\mathbf{y}_j)$ as a density in the case where \mathbf{Y}_j is discrete valued due to being a counting measure, for example, number of crimes in a year. (McLachlan & Peel 2000)

In finite mixture models, the number of sources of data ie. number of distributions is assumed to be finite, yet the number of components, k , can be unknown in some cases and has to be inferred from the available data (McLachlan & Peel 2000). An example of mixture distributions that is produced with two normal distributions can be seen in figure 2.1. The plot was made by simulating two normal distributions and forming a mixture distribution from them. The two normal distributions share the same standard deviation but different means and mixing proportions, or weights. The mixture distribution of the two resembles more the normal distribution with higher weight, correspondingly.

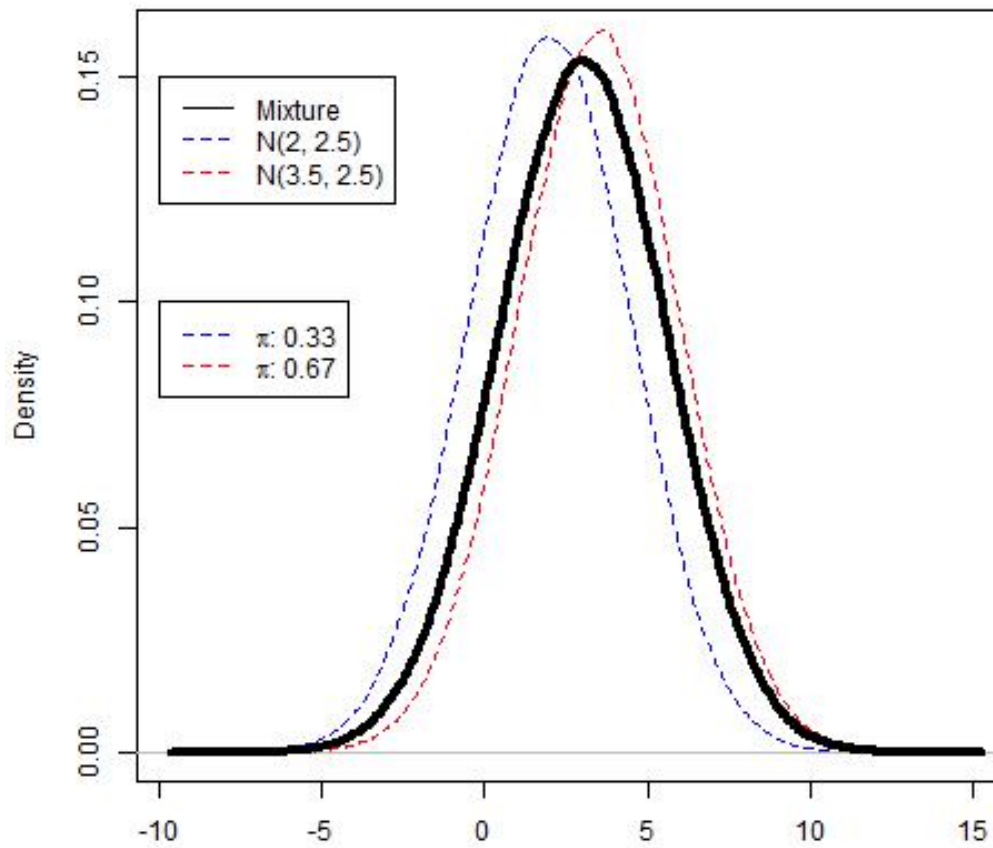


Figure 2.1. Simple mixture distribution and the distributions it is formed of.

In figure 2.1, the mixture distribution has similar bell-shape as a normal, or, Gaussian distribution but this is not always the case. If the distributions from which the mixture distribution is from the shape of the said density function can be very different. In figure 2.2 the mixture of three normals is very different from the bell curve.

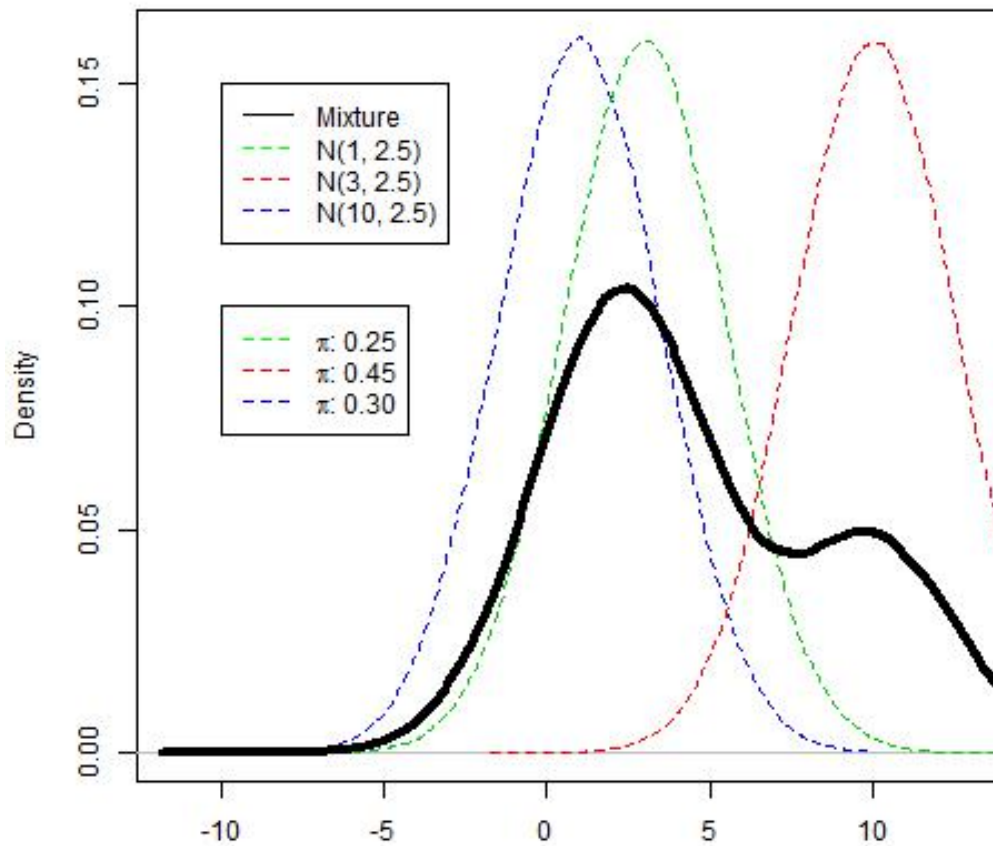


Figure 2.2. Mixture from three Gaussian distributions.

So far in this thesis, only one-dimensional cases of the mixture models have been described. In practice, one dimension means a single variable in the data set, say, the height of an individual. If the characteristic of interest was indeed the height of an individual, it makes sense to regard the overall distribution of a population as a mixture of multiple normal distributions, since in every age group (eg. children, adolescents, and adults) the height measurements are probably more or less normally distributed. However, mixture models can be extended to multidimensional space, which means that they can also be utilized for clustering purposes in multivariate situations. Next chapter explores the *Expectation Maximization* algorithm and maximum likelihood fitting which are crucial parts in forming the clusters in a mixture modeling framework.

2.2.1 Maximum likelihood fitting and EM algorithm

Maximum likelihood (ML) methods are commonly used methods for fitting finite mixture models. In short, the goal of ML estimation is to calculate an estimate for each set of n data items, so that the likelihood function is maximized for the obtained parameter estimate (McLachlan & Peel 2000). This estimate can be expressed as $\hat{\Psi}$, where vector Ψ is a d -dimensional parameter vector in density function $f(y_j; \Psi)$ and $\hat{\Psi}$ is the estimate based on these n data.

The aforementioned estimate $\hat{\Psi}$ is obtained in regular situations by an appropriate solution of the likelihood equation,

$$(2.5) \quad \partial L(\Psi)/\partial \Psi = \mathbf{0},$$

or,

$$(2.6) \quad \partial \log L(\Psi)/\partial \Psi = \mathbf{0},$$

where

$$(2.7) \quad L(\Psi) = \prod_{j=1}^n f(y_j; \Psi)$$

denotes the likelihood function for Ψ formed assuming independent vectors y_1, \dots, y_n . Maximizing the log-likelihood function directly is extremely complicated since it requires optimization for multiple parameters. Hence, the maximum likelihood estimator (MLE) is usually obtained with the Expectation Maximization (EM) algorithm in cases of finite mixture models. (McLachlan & Peel 2000, chapter 2)

The EM-algorithm, proposed by Dempster et al. (1977), is an iterative process that has been shown to be very useful in finding a local maximizer for the likelihood function. In the framework of EM, the data are being regarded as being incomplete because of group indicator vectors z_1, \dots, z_n that are not directly observed. In other words, each z_j indicate group-membership in a component of the mixture model 2.2 with $z_{ij} = (z_j)_i = 1$ or 0, telling whether y_j arose from the i th component of the mixture or not.

The complete-data vector is expressed as $y = (y_i^T, z_i^T)^T$. Due to this incomplete-data structure, z_1, \dots, z_n are taken to be realized values of the random vectors Z_1, \dots, Z_n , where they can be assumed to be distributed as

$$(2.8) \quad Z_1, \dots, Z_n \sim Mult_k(1, \pi)$$

(McLachlan & Peel 2000, chapter 2)

The multinormal assumption thus means that the distribution of Y_c , or the complete-data vector, implies the distribution for the incomplete-data vector Y . Now, the complete-data log likelihood for Ψ is given by

$$(2.9) \quad \log L_c(\Psi) = \sum_{i=1}^k \sum_{j=1}^n z_{ij} \{\log \pi_i + \log f_i(\mathbf{y}_j; \boldsymbol{\theta}_i)\},$$

where z_{ij} is treated as missing data. (McLachlan & Peel 2000, chapter 2)

The EM algorithm is comprised of two "steps": *Expectation* step and *Maximization* step. In the Expectation step (E-step), $Q(\Psi; \Psi^{(t)})$ is defined as the expected value of the complete-data log-likelihood function of Ψ , that is

$$(2.10) \quad Q(\Psi; \Psi^{(t)}) = E_{\Psi^{(t)}}[\log L_c(\Psi|\mathbf{y})],$$

where $\Psi^{(t)}$ is the value for Ψ at the previous iteration t and $\Psi^{(0)}$ would denote the initial value. The E-step on the iteration $t + 1$ needs only the current computation of the present conditional expectation of Z_{ij} given the observed data \mathbf{y} , where Z_{ij} is the variable denoting whether sample y_j is from component i . Thus, the posterior probability of y_j belonging to the i th component of the mixture on the t th iteration one can obtain with

$$(2.11) \quad E_{\Psi^{(t)}}(Z_{ij}|\mathbf{y}) = pr_{\Psi^{(t)}}\{Z_{ij} = 1|\mathbf{y}\} = \tau_i(\mathbf{y}_j; \Psi^{(t)})$$

where,

$$(2.12) \quad \tau_i(\mathbf{y}_j; \Psi^{(t)}) = \pi_i^{(t)} f_i(\mathbf{y}_j; \boldsymbol{\theta}_i^{(t)}) / \sum_{h=1}^k \pi_h^{(t)} f_h(\mathbf{y}_j; \boldsymbol{\theta}_h^{(t)}).$$

Now, with equation 2.11 one has

$$(2.13) \quad Q(\Psi; \Psi^{(t)}) = \sum_{i=1}^k \sum_{j=1}^n \tau_i(\mathbf{y}_j; \Psi^{(t)}) \{\log \pi_i + \log f_i(\mathbf{y}_j; \boldsymbol{\theta}_i)\}$$

on taking the conditional expectation of complete-data log-likelihood for Ψ given \mathbf{y} . (McLachlan & Peel 2000, chapter 2)

Now, when considering the $(t + 1)$ th iteration and having the updated estimate $\Psi^{(t+1)}$, the M-step demands the global maximization of the aforementioned $Q(\Psi; \Psi^{(t)})$ with respect to Ψ over Ω , the parameter space (McLachlan & Peel 2000, chapter 2). In short, the M-step can be written as

$$(2.14) \quad Q^{t+1} = \operatorname{argmax} Q(\Psi|\Psi^{(t)}).$$

and the EM-algorithm generally means repeating steps 2.13 and 2.14 iteratively until convergence ie. until the quantity Q^{t+1} no longer changes after M-step.

With a finite mixture model, the new estimates of the mixing weights π_i^{t+1} are computed independently of the new estimate Ψ^{t+1} of the parameter vector Ψ that contains the unknown parameters in the component densities. If the z_{ij} were not unknown but observable, the complete-data MLE of π_i would be simply

$$(2.15) \quad \hat{\pi}_i = \sum_{j=1}^n z_{ij}/n \quad (i = 1, \dots, k).$$

Should z_{ij} be unknown, they will simply be replaced with their respective current conditional expectation $\tau_i(y_j; \Psi^{(t)})$ and so the new estimate of π_i is given by

$$(2.16) \quad \hat{\pi}_i = \sum_{j=1}^n \tau_i(y_j; \Psi^{(t)})/n \quad (i = 1, \dots, k).$$

(McLachlan & Peel 2000, chapter 2)

2.2.2 Multivariate normal mixtures

When considering model-based clustering, every component of a mixture density is often connected to a cluster. A Gaussian mixture model assumes a normal distribution for every component, which means that the features of the cluster distributions, such as volume, shape and orientation of the cluster i are determined by the covariance matrix Σ_i , and the clusters are ellipsoidal and centered at the mean vector μ_i . (Scrucca et al. 2017)

An eigenvalue decomposition of these group covariance matrices yield covariance structures that can be used for finding groups from the data. The work of Banfield & Raftery (1993), Celeux & Govaert (1995), and Fraley & Raftery (2002) eventually inspired the creation of widely used R package *mclust*, which is the primary tool for analyzing the real world data in this thesis. (McNicholas & Murphy 2010)

For identifying models, letter abbreviations are used for describing the nature of the *mclust* -family models and clusters. There are three letters used here in the abbreviations, and the possible options for the letter are E,V, and I. In the case of the first letter, and in one dimensional context, E stands for equal variance for each cluster, V stands for varying variance over clusters, and the letter I refers to the identity matrix here meaning unit variance. Thus, in one dimension, there would be only two model possibilities with either equal (EII) or varying (VII) variance. In the multivariate case, 14 models can be specified each having different geometric characteristics. Shape, volume, and orientation of the covariance matrix can be set to be equal or variable across clusters, and these possible models are listed in table 2.1. In table 2.1 one can also see different parametrizations of the covariance matrix of the k th cluster that is based on the eigenvalue decomposition $\Sigma_k = c_i \mathbf{T}_k \Lambda_i \mathbf{T}_k'$, where c_i is a scaling constant, \mathbf{T}_i is a matrix of eigenvectors, and Λ_i is a diagonal matrix of scaled eigenvalues. (Scrucca et al. 2017)

Table 2.1. Different parametrizations in accordance with Scrucca et al. (2017)

Identifier	Model	Distribution	Volume	Shape
EII	cI	Spherical	Equal	Equal
VII	$c_i I$	Spherical	Variable	Equal
EEI	$c\Lambda$	Diagonal	Equal	Equal
VEI	$c_i \Lambda$	Diagonal	Variable	Equal
EVI	$c\Lambda_i$	Diagonal	Equal	Variable
VVI	$c_i \Lambda_i$	Diagonal	Variable	Variable
EEE	$cT\Lambda T'$	Ellipsoidal	Equal	Equal
EVE	$cT\Lambda_i T'$	Ellipsoidal	Equal	Variable
VEE	$c_i T\Lambda T'$	Ellipsoidal	Variable	Equal
VVE	$c_i T\Lambda_i T'$	Ellipsoidal	Variable	Variable
EEV	$cT_i \Lambda T'_i$	Ellipsoidal	Equal	Equal
VEV	$c_i T_i \Lambda T'_i$	Ellipsoidal	Variable	Equal
EVV	$cT_i \Lambda_i T'_i$	Ellipsoidal	Equal	Variable
VVV	$c_k T_i \Lambda_i T'_i$	Ellipsoidal	Variable	Variable

To clarify, let's examine an example of a multivariate model VEE: the three letter identifier indicates that the clusters have varying variances (V), equal ellipsoidal distributions (E), and equal shapes (E). A visual presentation of such model, along with other possibilities shown in table 2.1 can be seen in figure 2.3. The visualisation of the geometric characteristics was presented in a paper by Scrucca et al. 2017.

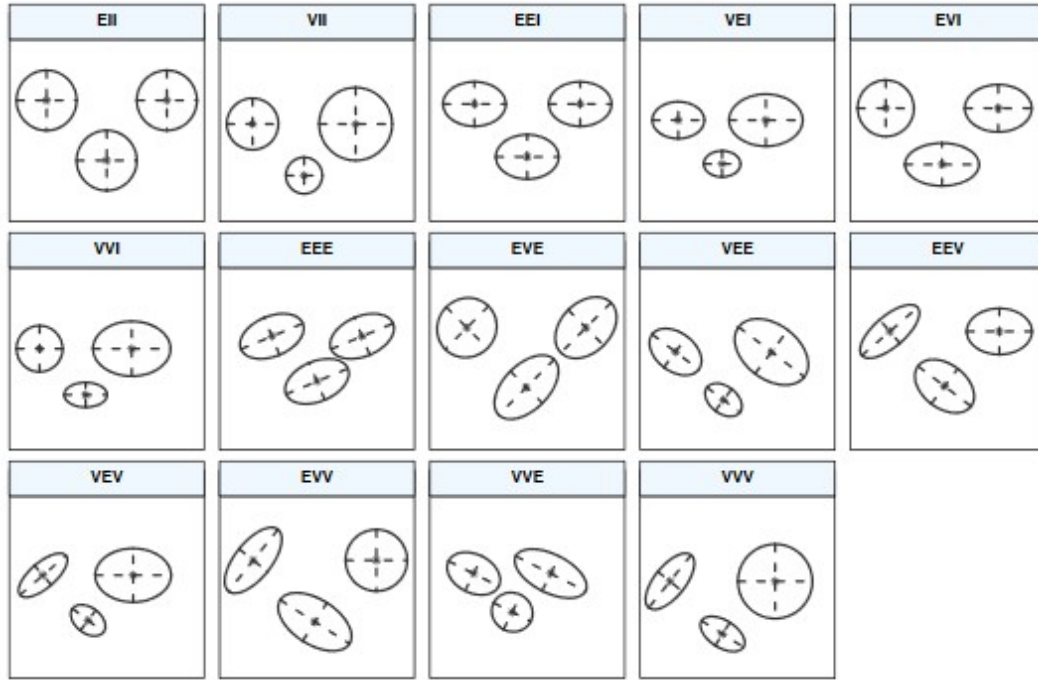


Figure 2.3. Geometric characteristics of multivariate mixture models (from paper by Scrucca et al. (2017). Permission for the usage of the image granted by Luca Scrucca).

From these models, the estimation of VEE and EVV models is conducted by the methods explained by Celeux & Govaert (1995), and models EVE and VVE are estimated with the approach described by Browne & McNicholas (2014). With models VEE, EVE and VVE there is an assumption that the mixture components have the same orientation matrix. This assumption is useful since because of that, a parameter sparing characterization of the clusters is possible without risking flexibility in shape and volume definition (Scrucca et al. 2017).

2.2.3 Model selection

As previously stated, the number of components of a mixture model may not always be known and has to be inferred from the data. Choosing the correct number of components is not always very easy but there are some ways for determining the best number of components. According to McLachlan & Peel (2000), two main purposes for mixture modeling are to provide a good semi-parametric framework in which to model unknown distributional shapes, and model-based clustering. From these two purposes, this thesis focuses on clustering and thus the ways to assess the number of components is presented with clustering in mind.

Some often used measures for assessing the number of components in mixture models are Akaike's Information Criterion (AIC) and Bayesian Information Criterion (BIC). From these two, BIC is often the default option for many tools, such as the *mclust* package in R and is in this thesis preferred over AIC. Being an information

criterion, BIC does not only address the issue of the number of components but also the model selection in multivariate normal mixture models. Information criteria are based on penalized likelihood and thus, as more components are added for the same fixed data, the likelihood after successful optimization increases and a penalty term for the number of estimated parameters needs to be subtracted from the log-likelihood. The BIC generally takes the form

$$(2.17) \quad BIC_{M,K} = -2 \log L(\hat{\Psi}) + d \log n$$

where d is the number of estimated parameters, M is the model (in multivariate normal mixtures context), and G is the number of components for which the log-likelihood is estimated. This form of BIC leads to a model being selected that minimizes BIC. Yet, in *mclust* the BIC is in the form

$$(2.18) \quad BIC_{M,k} = 2 \log L(\hat{\Psi}) - d \log n$$

that in turn indicates a model is selected that maximizes BIC. (McLachlan & Peel 2000, chapter 6; Scrucca et al. 2017)

However, BIC has a tendency of choosing the number of mixture components for approximating the density, rather than the number of underlying clusters. Because of this, Biernacki et al. (2000) have proposed other criterion for model selection in multivariate normal mixture models that would be more suitable for choosing the number of clusters: the integrated complete-data likelihood (ICL) criterion. (Scrucca et al. 2017)

ICL is similar to BIC but it penalizes BIC through an entropy term which measures the overlap of clusters, and it has been shown to yield good results if there is not much overlap. ICL can be defined by

$$(2.19) \quad ICL_{M,k} = BIC_{M,k} + 2 \sum_{j=1}^n \sum_{k=1}^K z_{ij} \log(p_{ij}),$$

where p_{ij} is the conditional probability that y_j comes from the i th component of the mixture model, and z_{ij} equals one if the j th sample is assigned to cluster i and zero otherwise. (Scrucca et al. 2017)

There is a great variety of other methods for model selection that are suitable for different situations and many of these methods are explored in detail by McLachlan and Peel (2000). In this thesis, in addition to BIC and ICL, likelihood ratio testing (LRT) is also used as tool for choosing a proper number of components for a model if an incidence occurs where optimal number of components cannot be determined by ICL and BIC.

Likelihood ratio test is conducted for testing supposed null hypothesis $H_0 : k = k_0$ against the alternative hypothesis $H_1 : k = k_1$ where $k_1 > k_0$. Usually,

$k_1 = k_0 + 1$ since common practice dictates that components are to be added one by one. Likelihood ratio test statistic (LRTS) is given by

$$(2.20) \quad LRTS = -2\log\{L(\hat{\Psi}_{k_0})/L(\hat{\Psi}_{k_1})\}$$

and the bigger the LRTS value is, the more evidence there is against H_0 (Scrucca et al. 2017). LRT significance is usually achieved by resampling methods. One suggested method for obtaining the null distribution of LRT and the p -value is bootstrapping. (McLachlan & Peel 2000; McLachlan 1987)

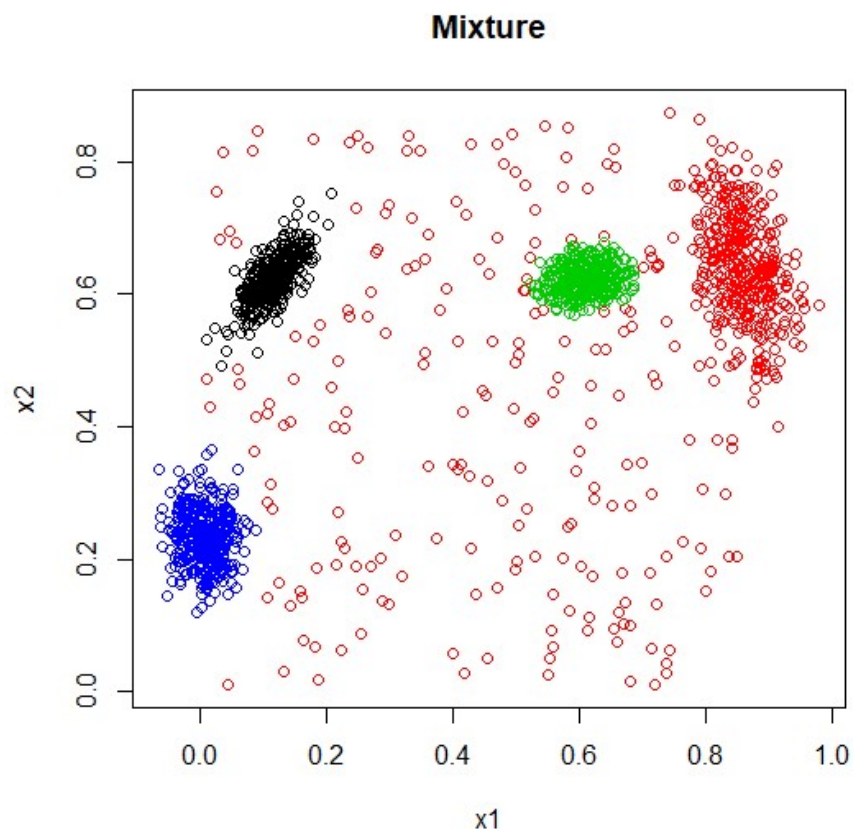
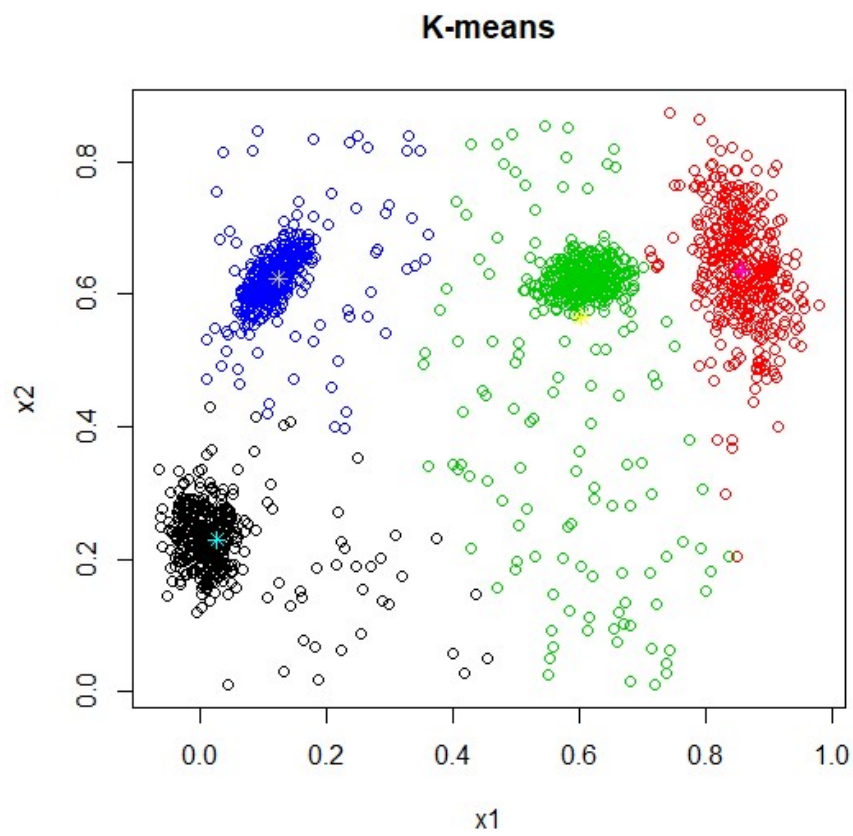
In the bootstrapping procedure, a bootstrap sample is first generated from the fitted mixture model with G_0 components. Then, LRTS is computed for the sample after fitting mixture models with G_0 and G_1 components. The bootstrap null distribution of LRTS is achieved by iterating previous phases multiple times (eg. 999 times) and p -value may then be computed:

$$(2.21) \quad p \approx \frac{1 + \sum_{b=1}^B I(LRTS_b \geq LRTS_{obs})}{B + 1}$$

where $LRTS_{obs}$ is the test statistic which is calculated on the sample, the indicator function that is equal to one if its argument is true and zero otherwise is denoted with $I()$, and B means the number of iterations. (Scrucca et al. 2017)

2.2.4 Practical example of the theoretical differences between the methods

Mixture models and k-means clustering are similar in a sense that they both function iteratively but they differ in terms of forming the clusters. This difference can be demonstrated with a visual example. For the example, a four-cluster data set with outliers is created using the *MixSim* (see Maitra & Melnykov (2010)) package in R. The difference between the methods is shown in figure 2.4. From these pictures it is possible to see the fundamental difference between k-means clustering and multivariate mixture model clustering in terms of cluster forming. K-means classifies outliers to the cluster whose centroid is closest to them and mixture models to a cluster that has the most similar structure. In the figure of k-means, the outliers are quite evenly divided into four clusters, whereas in figure of mixture model clustering the red cluster has its data points more scattered than the other clusters and thus, the mixture model "sees" outliers as part of the structure of the biggest cluster.



20

Figure 2.4. Cluster partition by k-means clustering and mixture models.

2.3 Clustering of longitudinal data

2.3.1 Mixture modeling in longitudinal data

In mixture modeling, like in many other statistical methods, there is an assumption of independence between observations. However, this usually cannot be assumed when a time aspect is involved, eg. in time series data or in longitudinal studies. There is likely to be some dependence between consecutive time points and thus, the assumption of independence is not fulfilled and the results may not be very reliable, robust or close to truth. There are ways, however, to deal with this issue in the framework of finite mixture models.

A special case that is derived from mixture models to assess the membership in a group through time is Group-Based Trajectory Modeling of development (GBTM), that has been pioneered by Nagin (2005) over the years. The goal of GBTM is to identify groups or clusters of individuals with similar trajectories in regards to some characteristic, and the model's estimated parameters are a result of maximum likelihood estimation (Nagin 2005). Thus, they are consistent and asymptotically normally distributed which are some of the wanted qualities of maximum likelihood parameter estimates as suggested by Cram r (1946) and also later in the field of econometrics by Thiel (1971) and Greene (1990).

This special case of mixture models is presented in a latent class regression framework which enables the use of covariates to identify cluster membership as well as the handling of the time element (Leisch 2003). The *mclust* family of models relies on the multivariate Gaussian structure of repeated measured data: each time point corresponds to a normal vector and it is possible to treat repeated measurements of a variable as a multivariate normal matrix. This way, *Mclust* models are able to perform longitudinal clustering as well. However, McNicholas & Murphy (2010) pointed out that the *Mclust* covariance structure is not very natural for the longitudinal correlation structure, and suggested a family of models of their own. These models, they claim, are more natural for longitudinal data and can also provide more information about the covariance structure and the autoregressive structure of longitudinal data.

In the suggested model for longitudinal data by McNicholas & Murphy, a Gaussian mixture is assumed for every component with a modified Cholesky-decomposed covariance structure (McNicholas & Murphy 2010). In this thesis, however, the clustering of longitudinal data is carried out using the multivariate normal assumption of each time point and modified Cholesky-decomposed covariance structure is not utilized due to the belief that modeling covariance structure is sufficient for taking the autocorrelation into account in the framework of this thesis.

2.3.2 KmL: k-means clustering for longitudinal data

KmL, or k-means clustering for longitudinal data, is an R package developed to be a non-parametric alternative for clustering longitudinal data. Its principle comes from k-means clustering: assign a center to a cluster then according to distance assign each point to its nearest cluster. *KmL* introduces the time aspect to this framework.

Let's consider a set S of n data points. For each data point, an outcome variable Y is measured at t different times. The value of Y for data point j at time l is noted as y_{jl} . For data point j , the trajectory is notated $\mathbf{y}_j = (y_{j1}, y_{j2}, \dots, y_{jt})$. The idea is to divide S into k homogeneous sub-groups. The distance metric in *KmL* is also usually Euclidean, but other distance metrics, such as Manhattan distance, are possible as well. (Genolini & Falissard 2010)

For choosing the optimal number of clusters, *KmL* algorithm uses the Calinski and Harabasz criterion which is denoted as $C(g)$ (Calinski & Harabasz 1974). The criterion utilizes two measures, between-variance and within-variance, to determine the optimal number of clusters. The between-variance is determined in the following way: let n_m be the number of trajectories in cluster m , $\bar{\mathbf{y}}_m$ the mean trajectory of cluster m , $\bar{\mathbf{y}}$ the mean trajectory of the whole set S and \mathbf{v}' denotes the transposition of vector \mathbf{v} . The between-variance matrix is

$$(2.22) \quad \mathbf{B} = \sum_{m=1}^k n_m (\bar{\mathbf{y}}_m - \bar{\mathbf{y}})(\bar{\mathbf{y}}_m - \bar{\mathbf{y}})',$$

the trace of the between-variance is obtained by summing the diagonal coefficients of the aforementioned matrix. If the between-variance is high the clusters are well separated, and if the between-variance is low, the clusters are close to each other. The within-variance is computed in similar fashion by denoting the within-variance matrix

$$(2.23) \quad \mathbf{W} = \sum_{m=1}^k \sum_{j=1}^{n_m} (\mathbf{y}_{mj} - \bar{\mathbf{y}})(\mathbf{y}_{mj} - \bar{\mathbf{y}})'.$$

Now, the actual criterion $C(g)$ is obtained by

$$(2.24) \quad C(g) = \frac{\text{Trace}(\mathbf{B})}{\text{Trace}(\mathbf{W})} \cdot \frac{n - k}{k - 1}.$$

(Genolini & Falissard 2010)

Low score of the within-variance means that the groups are compact whereas high score of the within-variance means that the groups are heterogenous.

2.4 Earlier comparisons between model-based clustering and k-means clustering

There has been comparisons between the two methods before. Baid et al. (2017) compared k-means and mixture models for brain tumor image data and in their study k-means did slightly better in this particular type of task. Steinley and Brusco (2011) stated that k-means clustering can perform as well as mixture model clustering based on their simulation studies. However, Vermunt (2011) provided a comment to that claim, stating that while it is true that k-means can perform as well as mixture

models in certain situations, it can also perform much worse, depending on the situation and the real world phenomenon. He provided his own experiments and theoretical background to this statement.

The idea of the superiority of mixture models over k-means is further enforced in a paper by Qiu (2010). The paper compares the classification performances of k-means clustering and mixture models in bivariate homoscedastic case. In the study, k-means performed poorly as the component distribution became more and more elongated, whereas mixture models could possibly take advantage of such change.

There is still evidence to suggest that k-means, or at least a k-means based method can do well against mixture models. Based on experiments, Genolini & Falissard (2010) noticed that their method did well in experiments against trajectory analysis: they suggested that their method could potentially outperform trajectory analysis in non-polynomial data. Moreover, their method, *KmL*, seems to be pioneering in a sense that it is specifically designed for longitudinal data. Still, they reflected that their method has weaknesses that any clustering method could have and recommended combined use of more than one method in clustering tasks.

3 Experiments on simulated data

In this chapter, the performance of the methods is discussed based on the experiments conducted on simulated data. All the data from simulations in this thesis are Gaussian and simulations were carried on using Python 3 language in Google Colab cloud environment. Library used for simulations is *scikit-learn* (Pedregosa et al. 2011). R's package *MixSim* (Maitra & Melnykov 2010) was considered for simulating the data but due to its time-complexity it was left out of these simulations.

Purity (3.1) is used as the evaluation metric for performance. Purity is an external evaluation metric which requires the information of the correct classes for the data for expressing the "correctness" of the clustering. Purity score is the percentage of how many data points were clustered correctly to the pre-determined, "ground truth" cluster.

When defining purity, it is assumed that one is given l categories, while the clustering method makes k clusters. The purity of the clustering is defined as

$$(3.1) \quad Purity = \frac{1}{N} \sum_{q=1}^k \max_{1 \leq j \leq l} N_q^j,$$

where N is the total number of samples and N_q^j is the number of samples in cluster q that belongs to original class j ($1 \leq j \leq l$). The purity measure was used as described by Kim & Park (2007).

3.1 Simulating the data

The experiments for evaluating the performance between the methods is carried out by four different approaches: 1) increasing the number of outliers, 2) increasing the average overlap of the clusters, 3) increasing the dimensionality of the data, and 4) increasing the number of observations in the data. There are two evaluation approaches to the simulations: cluster purity and the difference between true cluster centers and the parameter estimates. For evaluation with purity, the simulations are carried out multiple times and the mean of the purity from all iterations is calculated in order to adjust for possible random effects that the simulation process may have.

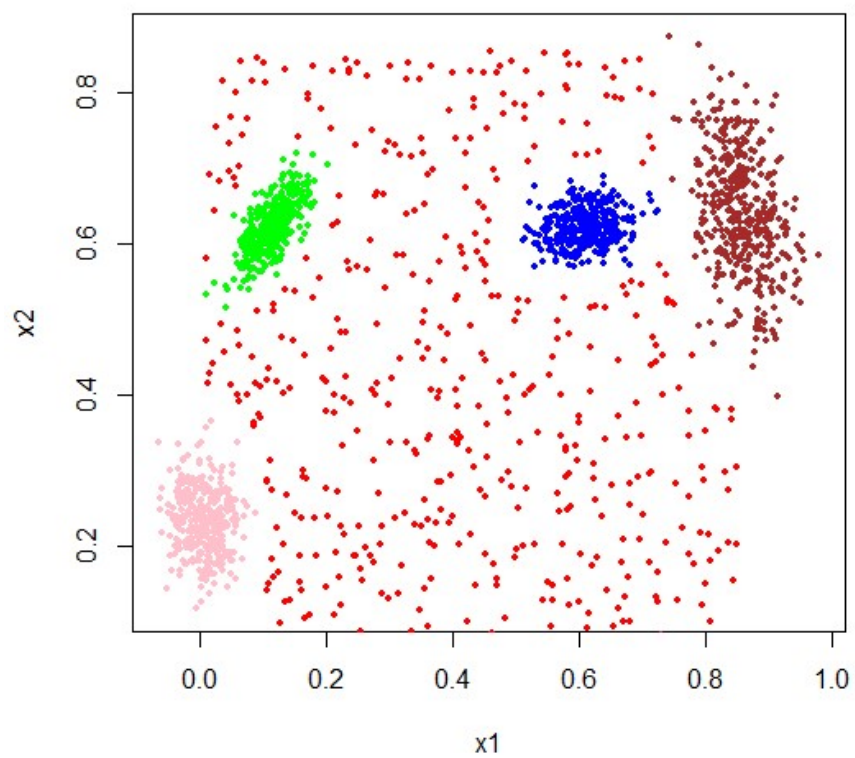
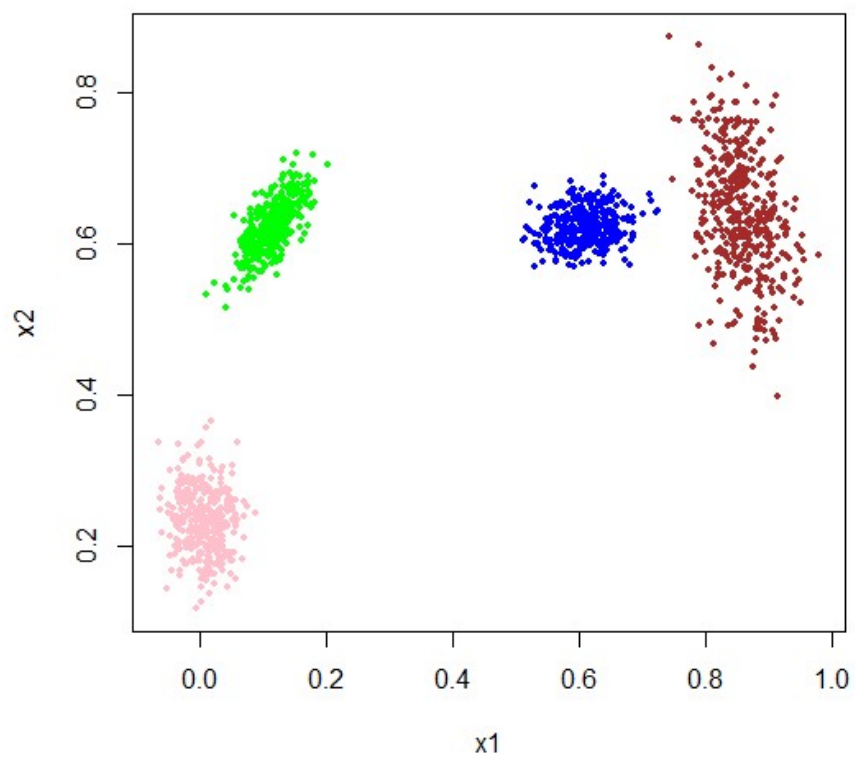
For evaluation with difference between cluster centers and parameter estimates, data sets are generated for each four approaches and the true centers of each cluster are computed. Then, estimates for these centers are calculated using mixture models and k-means, and the comparison is carried out by computing the Euclidean distance between the true centers and the estimates from the methods. The smaller the distance, the better the method is able to estimate the true center of the cluster.

Simulation procedures

1. *Increasing the number of observations:* Data sets are iteratively generated without outliers, without great overlap between the clusters (cluster standard deviation allowed to range from 0.01 to 0.05), and with dimensionality fixed to four dimensions. Only the number of observation is increased with each iteration.
2. *Increasing the overlap between the clusters:* A data set without outliers and with small overlap is created. For each iteration, a new data set with similar parameters as before is generated. Only the overlap between the clusters is increased by increasing cluster standard deviation. The dimensionality is fixed to two dimensions.
3. *Increasing the dimensionality:* A data set with two dimensions (ie. variables) is created in the first iteration. The same steps are repeated as in previous approaches but in each iteration a new dimension is added.
4. *Increasing number of outliers:* Data sets are iteratively generated and with each iteration the number of outliers is increased. The dimensionality of the data is fixed to two dimensions for each iteration. The possible effect on clustering performance should be caused by the outliers.

The number of clusters is assumed to be known and this information is provided for both methods. Thus, the goal is not to test how the suitable number of components is found but just how well a method performs in different situations. There are fewer iterations when the methods are compared for the Euclidean distance but the approaches are similar.

Example visualizations of simulations The simulated data can be visualized in 2-dimensional plots. An example of well separated clusters can be seen from figure 3.1. Figure 3.1 also visualizes the same example with added outliers. Figure 3.2 is an example of four clusters with increased overlap.



26
Figure 3.1. Data set with and without outliers.

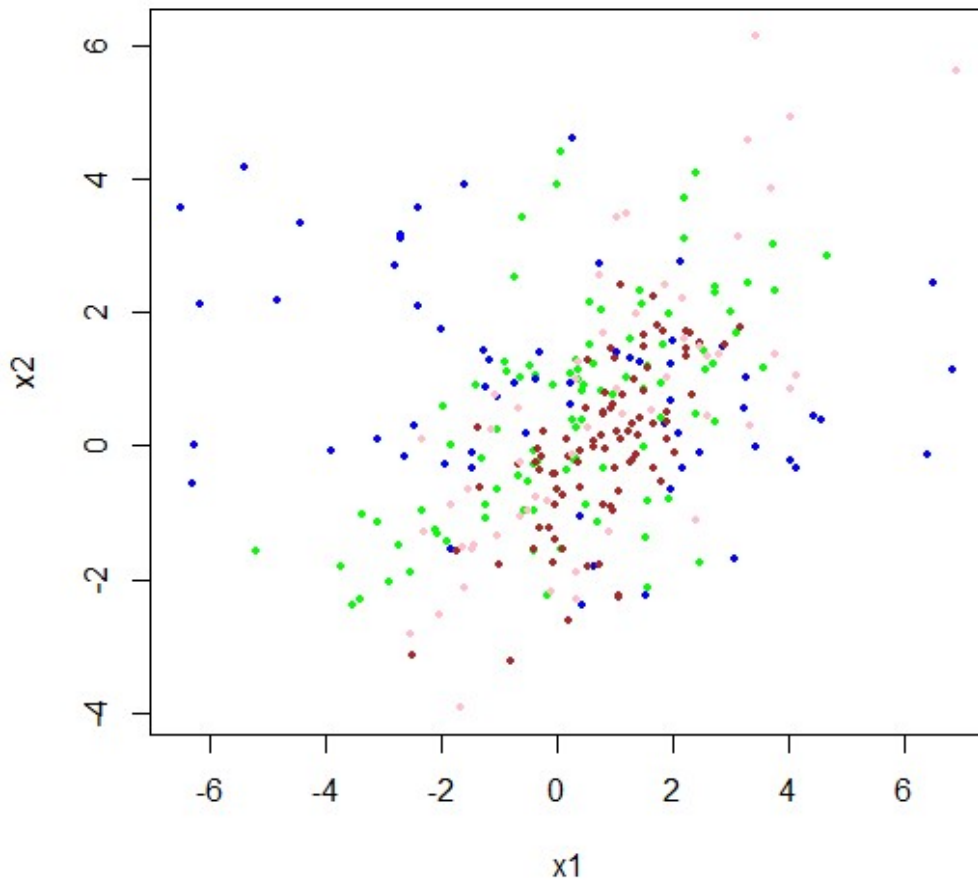


Figure 3.2. Four simulated clusters with 50% overlap.

3.1.1 Generating the data

Simulations are implemented using Python's Scikit-Learn library (Pedregosa et al. 2011) which has a ready-made function for generating Gaussian clusters. The clusters can be manipulated by altering the parameters of the function to fit the needs of these simulations.

3.2 Performance on simulated data

Both methods were tested on same data sets. Since k-means and mixture models require the information of the number of clusters or mixture components, the number of true clusters was fixed to four clusters. For mixture models, the initialization was done using k-means clustering. Although the initialization was done using k-means, the final attributes of the mixture model clustering are not restricted by the

initialization and thus k-means can be used for that purpose and the comparisons are still meaningful. Due to the existing randomness in the algorithms of the simulations the simulations were carried out 20 times and the mean of the purity score over all repetitions was calculated for obtaining the average performance of both methods.

For mixture models, there are four covariance structures that were used for modeling the simulated data. The structures are "full", "diagonal", "tied" and "spherical". "Full" structure means that each components has its own general covariance matrix, "diagonal" means that each component has its own diagonal covariance matrix, "tied" means that all components share the same general covariance matrix and "spherical" means that each component has its own single variance which is used for each dimension in a diagonal covariance matrix.

3.2.1 Performance in terms of cluster purity

Increasing number of observations The first simulation was done for the purpose of testing the effect of increasing observations. As the overlap between the clusters, determined by the cluster standard deviation, was set to be minimal, both methods performed well with increasing number of observations. The purity score didn't decrease below 100% at any point which is an expected result, since well separated clusters are easily divided into separate groups even by eye in two-dimensional space (see figure 3.12). In graph 3.3 every line is overlapping as both k-means and Gaussian mixture models with every one of its covariance structure alternatives has clustered all the data points perfectly.

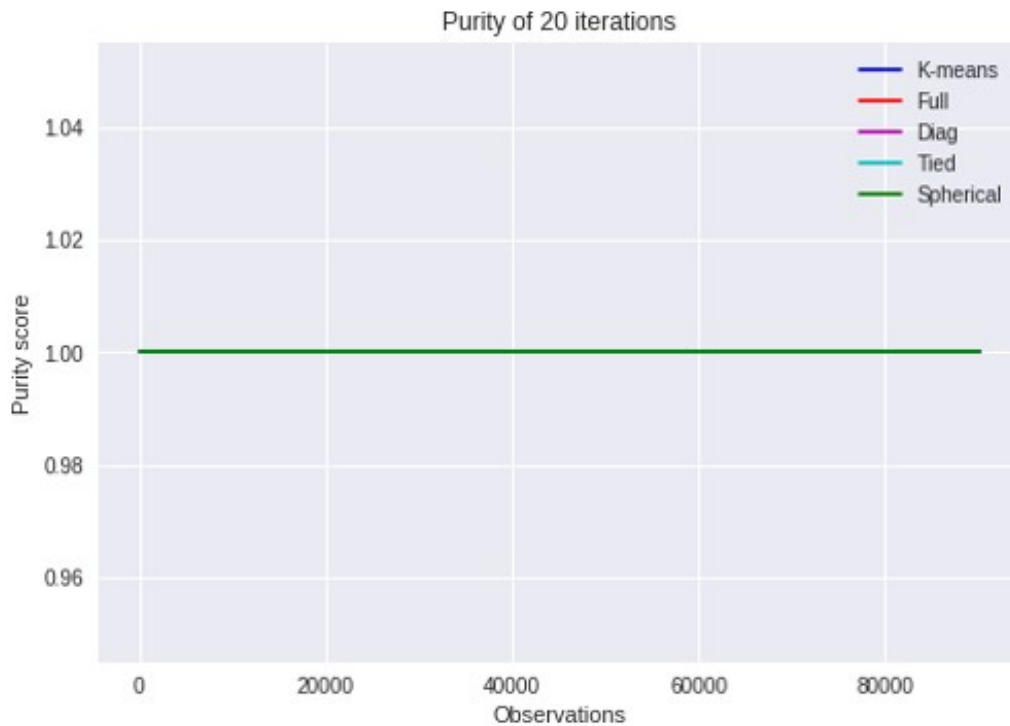


Figure 3.3. Performance with increasing number of observations

However, it is also possible that the k-means initialization of mixture models is a helpful factor for its good performance.

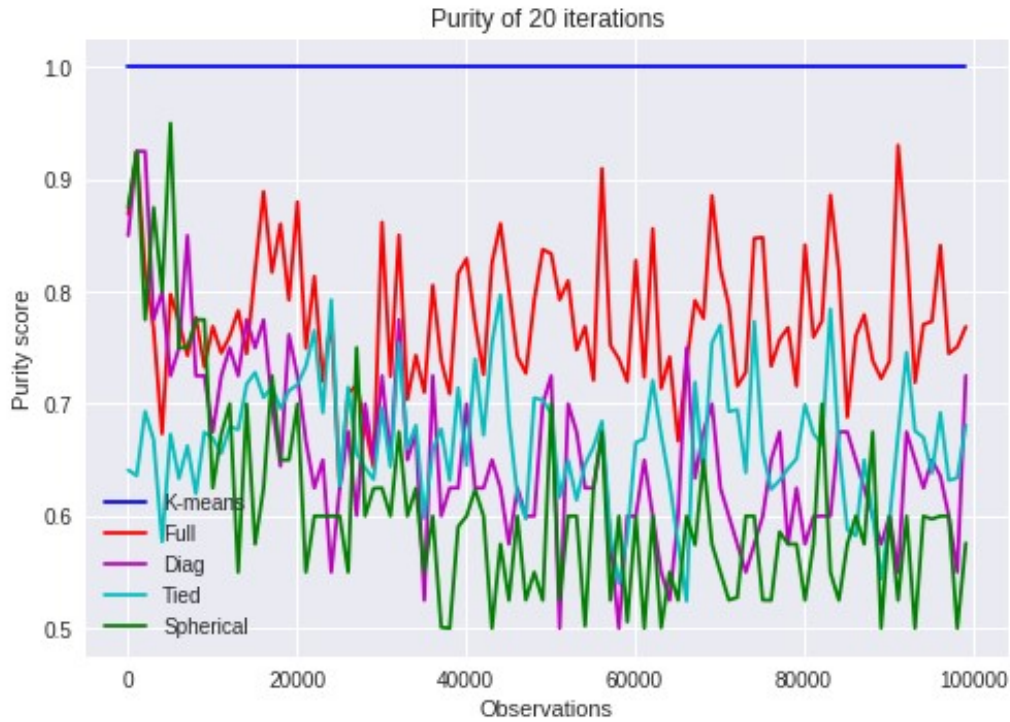


Figure 3.4. Performance with increasing observations, random initialization.

Graph 3.4 shows that if the same simulation is carried out with random initialization for Gaussian mixture models, they perform much worse than they did with k-means initialization. Still, this only means that mixture models benefit from initialization since random initialization that was used in 3.4 simulations puts mixture models into a disadvantage as initial values, responsibilities, and weights are given randomly. If mixture models were initialized with even remotely decent mixing weights and means for components, the performance was as good as with k-means. In the next simulations, mixture models are no longer initialized randomly.

Increasing overlap As the average overlap between the clusters was increased by increasing the standard deviation of the mixture components that generate the data, the performance in terms of purity score deteriorated with both methods. Figure 3.5 shows the development of purity score as cluster standard deviation increases from 0.1 to 1. In the picture the different covariance structures of Gaussian mixture models are presented in different colors, as is k-means clustering. Figure 3.6 shows the performance of each covariance structure separately with k-means and based on the pictures, it is not possible to state if there is any difference between the methods as the overlap between the clusters increases. In addition to that, the purity score is over 98% even with cluster standard deviation being 1. To further explore the effect of overlap between the clusters, cluster standard deviation is increased up to 10.

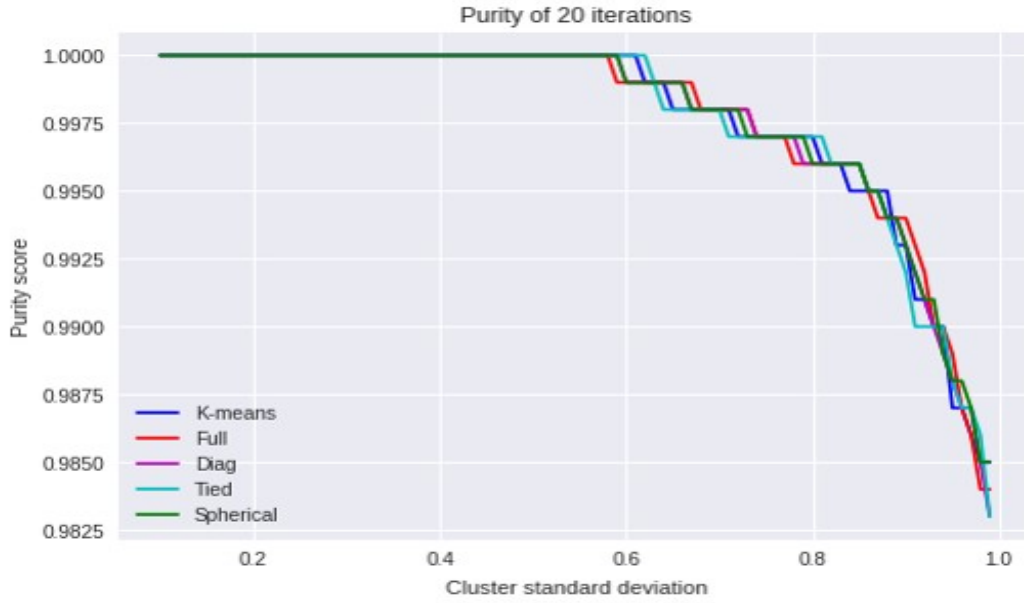


Figure 3.5. Performance with increasing overlap.

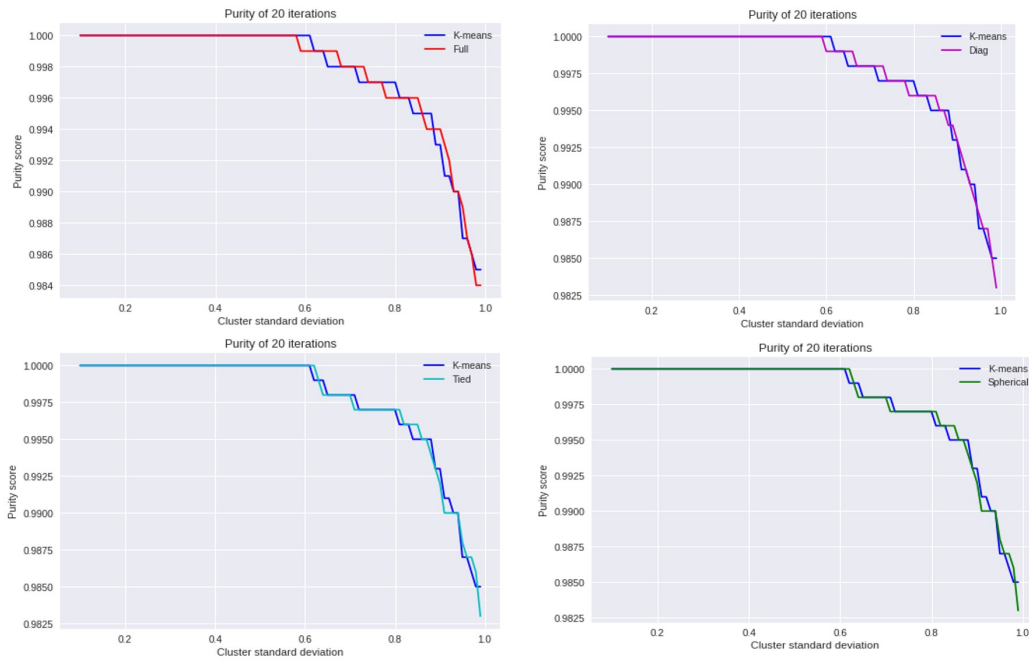


Figure 3.6. Performance with increasing overlap, separate plots for each covariance structure.

Figure 3.7 shows that the overlap affects both methods greatly, even to a point of purity being under 40%. From figures 3.7 and 3.8 it is possible to notice that up until the point where cluster standard deviation is 4, the purity is equal or slightly better with Gaussian mixture model clustering if full or tied covariance structure is used. However, with diagonal or spherical covariance structure the performance is

equal to k-means or little poorer. Both methods have very similar performance when standard deviation is above 4.

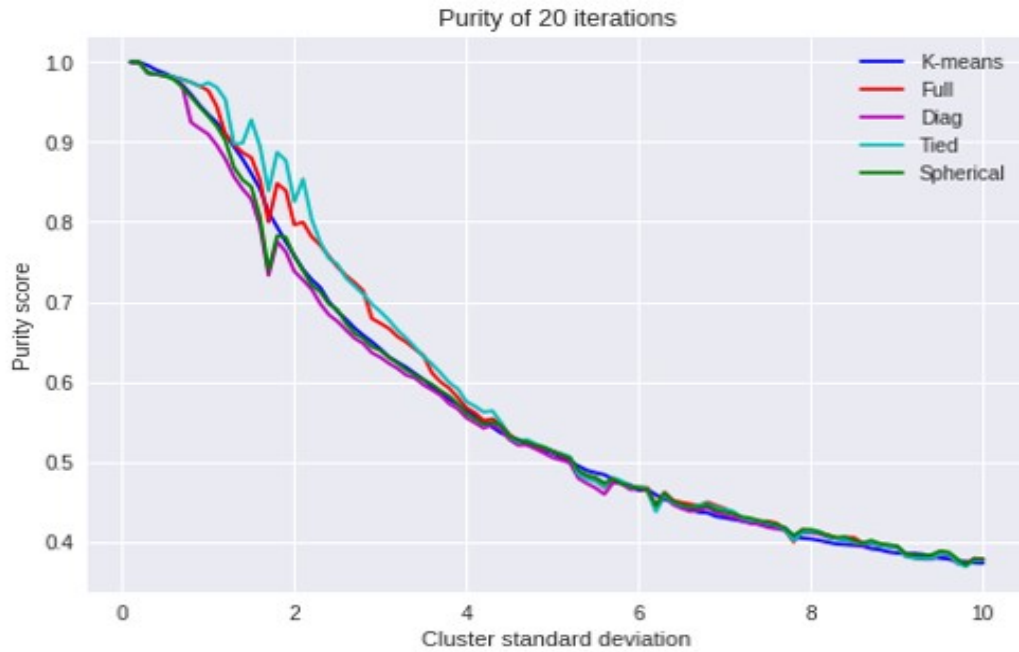


Figure 3.7. Performance with increasing overlap.

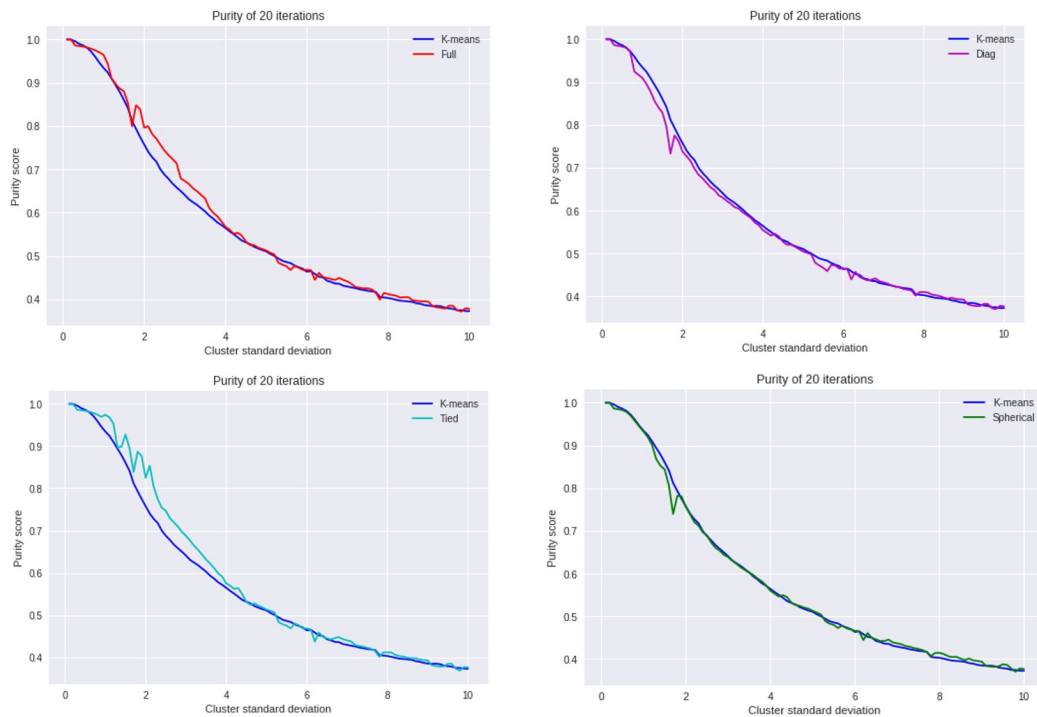


Figure 3.8. Performance with increasing overlap, separate plots for each co-variance structure.

Intuitively, it is more difficult to divide data points into separate clusters if clusters are very intertwined. Based on simulations conducted here, it is possible to suggest that with Gaussian mixture model clustering one may achieve slightly better results than with k-means clustering if the clusters overlap. This requires a proper covariance structure to be chosen for the data.

Increasing dimensionality Increase in dimensionality didn't have much effect on either of the methods, and although in graph 3.9 k-means seems to have some fluctuation in its performance, it seems to be random and probably due to random initialization. Also, the fluctuation is quite small as the purity never decreases below 90%.

Gaussian mixture models do not seem to be affected by the increase in dimensionality. This is curious, since the covariance structure is affected by the changes in dimensions of the data but it does not diminish the purity of Gaussian mixture models' performance.

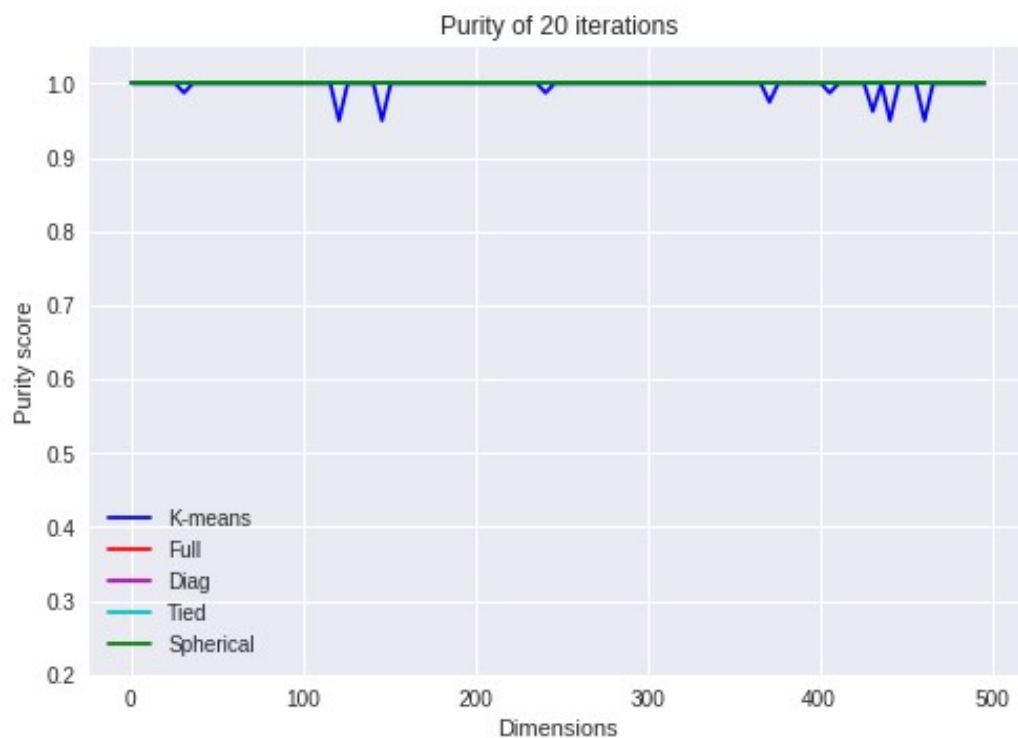


Figure 3.9. Performance with increasing dimensionality.

Increasing number of outliers As one could expect, increasing the number of outliers in the data resulted in decreasing purity score, regardless of the method. Gaussian mixture models had consistently higher purity scores as the number of outliers increased in the data set. The difference was not large but still noticeable.

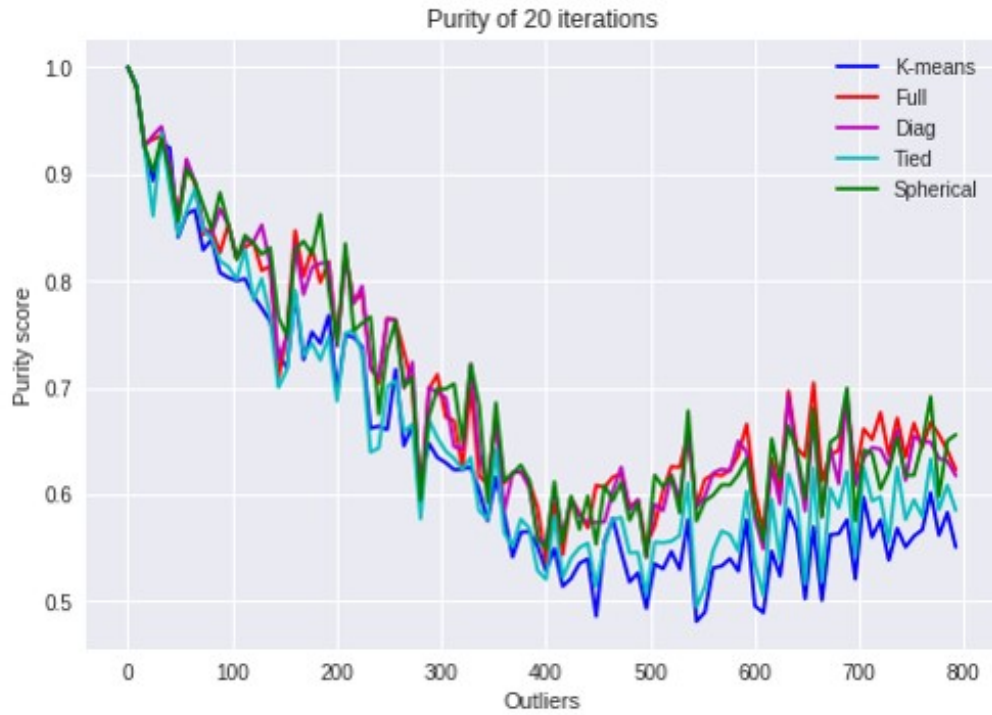


Figure 3.10. Performance with increasing number of outliers

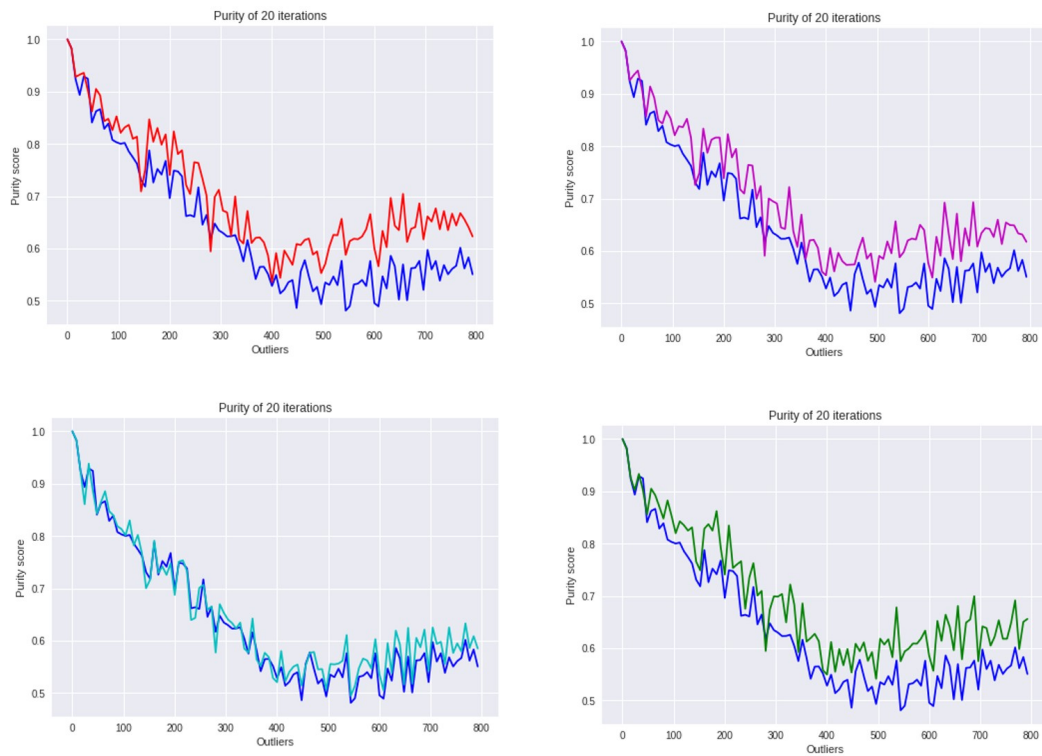


Figure 3.11. Performance with increasing number of outliers, separate plots for each covariance structure

Surprisingly, with both methods the purity score seemed to remain or even slightly increase after certain number of outliers was added to the data. This is because the outliers started to form the majority of the data so that the outliers cover the area of all the clusters, and as this area is divided to the fixed number of clusters the algorithms are told to find, the true data points are bound to be clustered to correct labels at least to some degree.

3.2.2 Performance in terms of distance between a parameter estimate and the true cluster center

Next, the methods are tested on the basis of how well they are able to estimate the cluster centers. Here, only the full covariance structure of mixture models is used as it was consistently superior compared to other covariance structures in previous experiments.

Increasing observations As observations were increased in simulated data sets, both methods were able to estimate the cluster center almost perfectly as one can see from figure 3.12. This is in accordance with previously obtained results where methods were evaluated in terms of cluster purity.

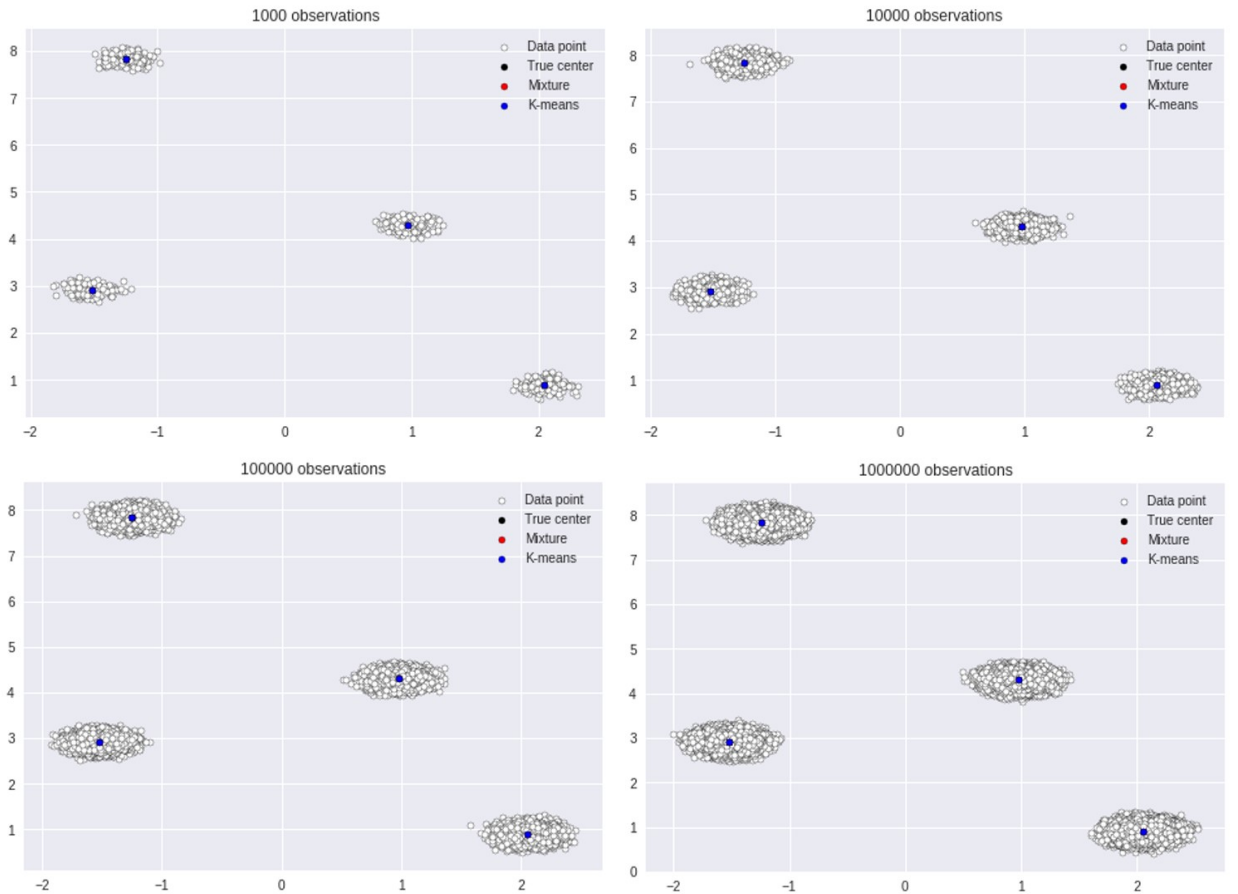


Figure 3.12. Cluster centers and parameter estimates.

Increasing overlap The differences between the methods are evident as the overlap is increased. With small overlap both methods manage to estimate the cluster centers well but as there is more overlap, mixture models seem to estimate the cluster centers closer to the true center.

Table 3.1. Euclidean distance of cluster centers and parameter estimates, cluster $sd = 1.1$

	Mixture	K-means
Cluster 1	0.07	0.15
Cluster 2	0.01	0.06
Cluster 3	0.01	0.09
Cluster 4	0.05	0.04

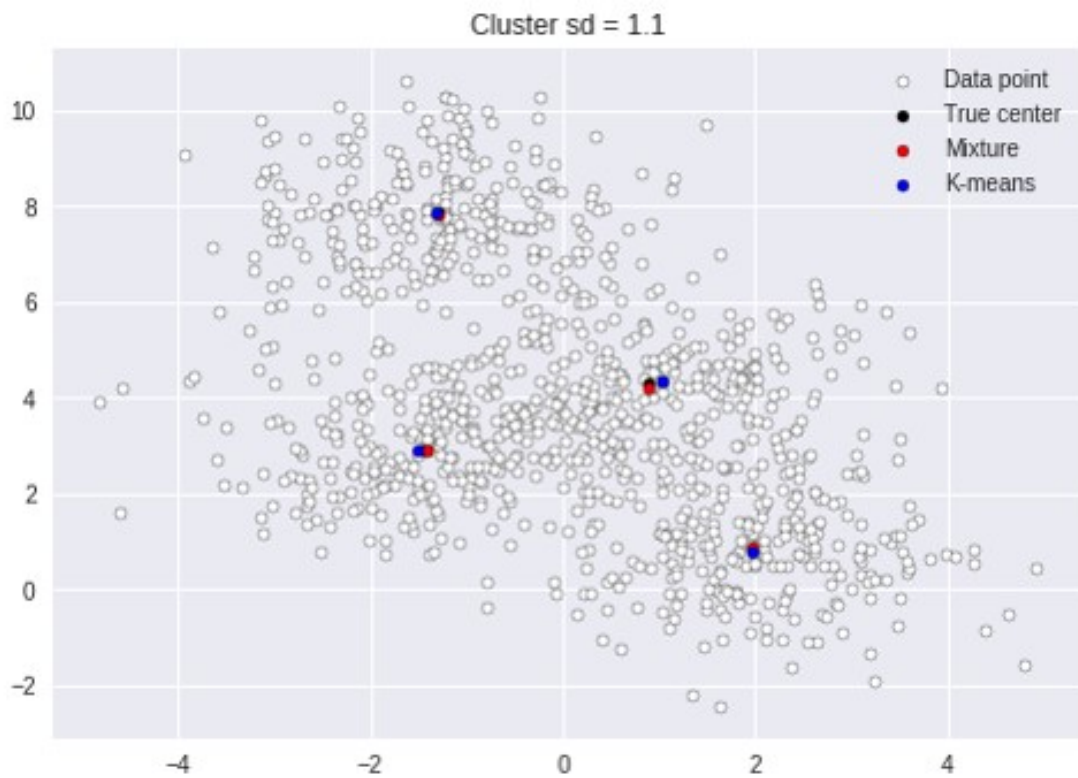


Figure 3.13. Cluster centers and parameter estimates.

Table 3.2. Euclidean distance of cluster centers and parameter estimates, cluster sd = 5

	Mixture	K-means
Cluster 1	2.49	3.82
Cluster 2	2.57	3.63
Cluster 3	1.94	3.42
Cluster 4	2.44	3.48

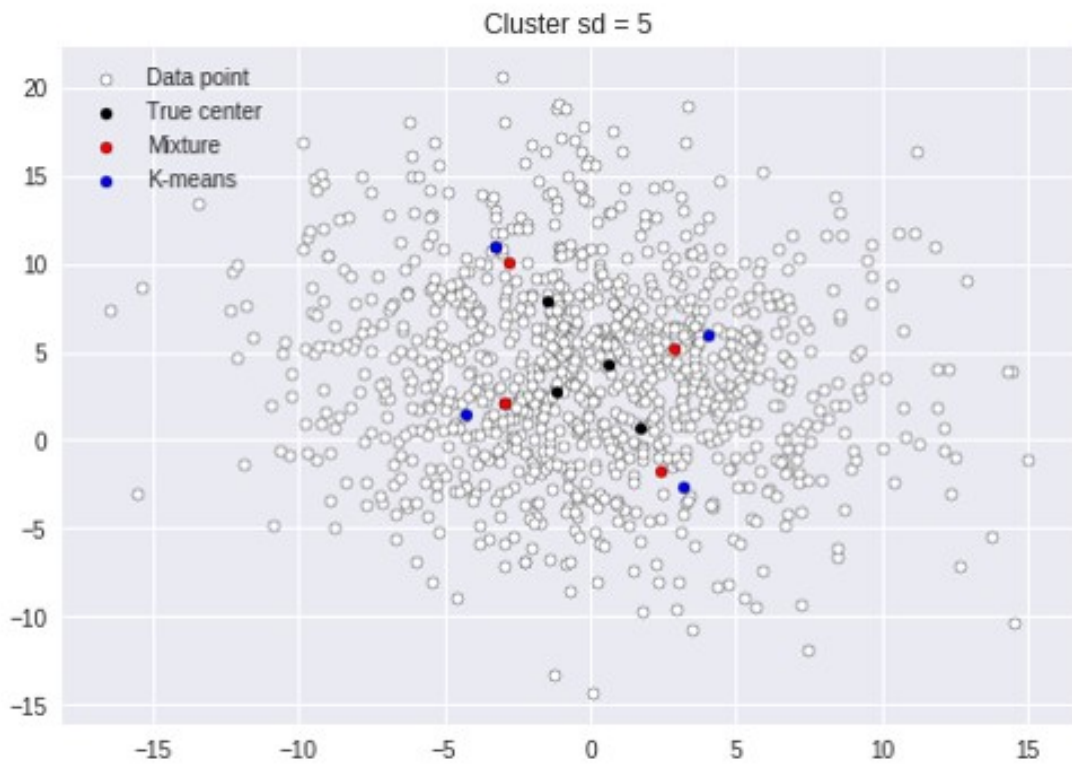


Figure 3.14. Cluster centers and parameter estimates.

Table 3.3. Euclidean distance of cluster centers and parameter estimates, cluster sd = 10

	Mixture	K-means
Cluster 1	5.01	8.37
Cluster 2	7.03	9.23
Cluster 3	6.70	8.80
Cluster 4	5.20	9.60

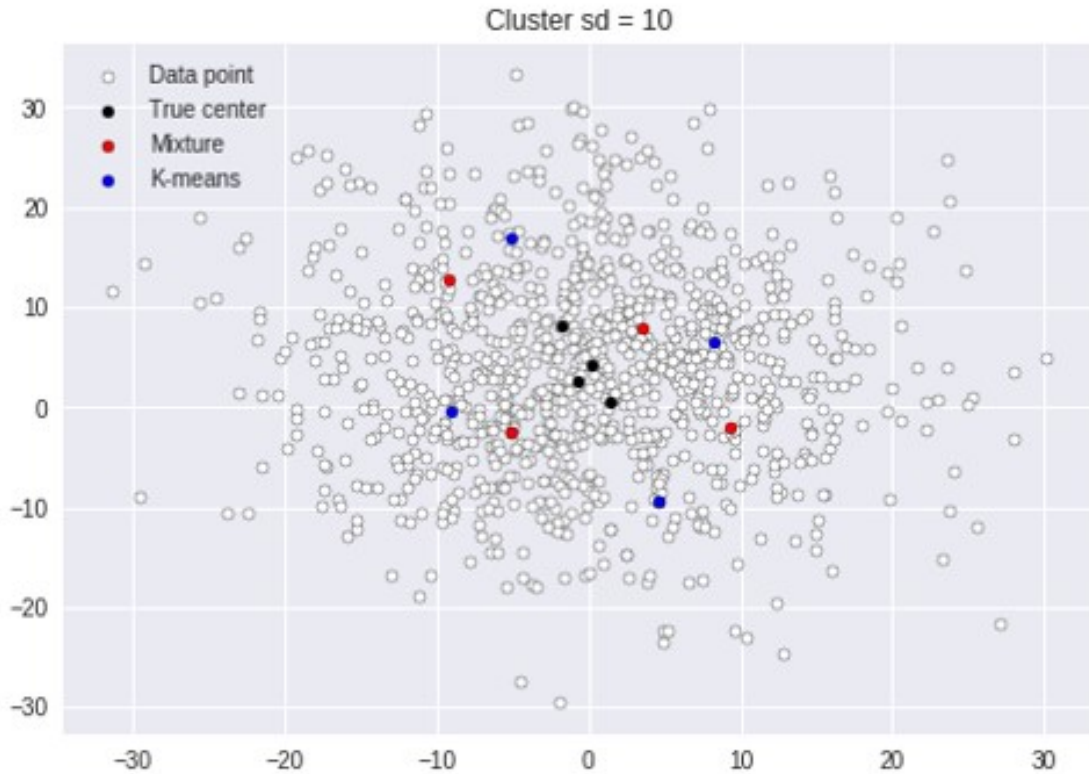


Figure 3.15. Cluster centers and parameter estimates.

Increasing dimensionality Increasing the number of dimensions in the data set didn't provide clear differences between the methods as both methods seemed to be able to estimate the multidimensional cluster centers rather well. There are differences between the parameter estimates and the true cluster centers, the Euclidean distance is very close to zero, as tables from ?? to ?? indicate. Though both methods get very close to the true values, mixture models seem to have slightly smaller distances between the estimates and the true values.

Table 3.4. Euclidean distance of cluster centers and parameter estimates with 2 dimensions

	Mixture	K-means
Cluster 1	$1.73 * 10^{-14}$	$1.53 * 10^{-14}$
Cluster 2	$7.49 * 10^{-15}$	$2.98 * 10^{-14}$
Cluster 3	$5.79 * 10^{-15}$	$9.92 * 10^{-15}$
Cluster 4	$1.03 * 10^{-13}$	$9.49 * 10^{-14}$

Table 3.5. Euclidean distance of cluster centers and parameter estimates with 20 dimensions

	Mixture	K-means
Cluster 1	$1.47 * 10^{-13}$	$1.20 * 10^{-13}$
Cluster 2	$6.99 * 10^{-14}$	$1.31 * 10^{-13}$
Cluster 3	$6.30 * 10^{-14}$	$1.46 * 10^{-13}$
Cluster 4	$1.38 * 10^{-13}$	$1.40 * 10^{-13}$

Table 3.6. Euclidean distance of cluster centers and parameter estimates with 200 dimensions

	Mixture	K-means
Cluster 1	$3.46 * 10^{-13}$	$4.47 * 10^{-13}$
Cluster 2	$3.43 * 10^{-13}$	$4.48 * 10^{-13}$
Cluster 3	$4.49 * 10^{-13}$	$4.49 * 10^{-13}$
Cluster 4	$3.74 * 10^{-13}$	$4.53 * 10^{-13}$

Table 3.7. Euclidean distance of cluster centers and parameter estimates with 2000 dimensions

	Mixture	K-means
Cluster 1	$1.14 * 10^{-12}$	$1.49 * 10^{-12}$
Cluster 2	$1.07 * 10^{-12}$	$1.48 * 10^{-12}$
Cluster 3	$1.11 * 10^{-12}$	$1.43 * 10^{-12}$
Cluster 4	$1.12 * 10^{-12}$	$1.49 * 10^{-12}$

Increasing outliers Increasing outliers provided similar results as the previous experiments as well: mixture models seemed to be able to estimate cluster center better than k-means clustering as the number of outliers was increased.

Table 3.8. Euclidean distance of cluster centers and parameter estimates with 10 outliers

	Mixture	K-means
Cluster 1	0.21	0.21
Cluster 2	0.15	0.15
Cluster 3	0.00	0.00
Cluster 4	0.06	0.06

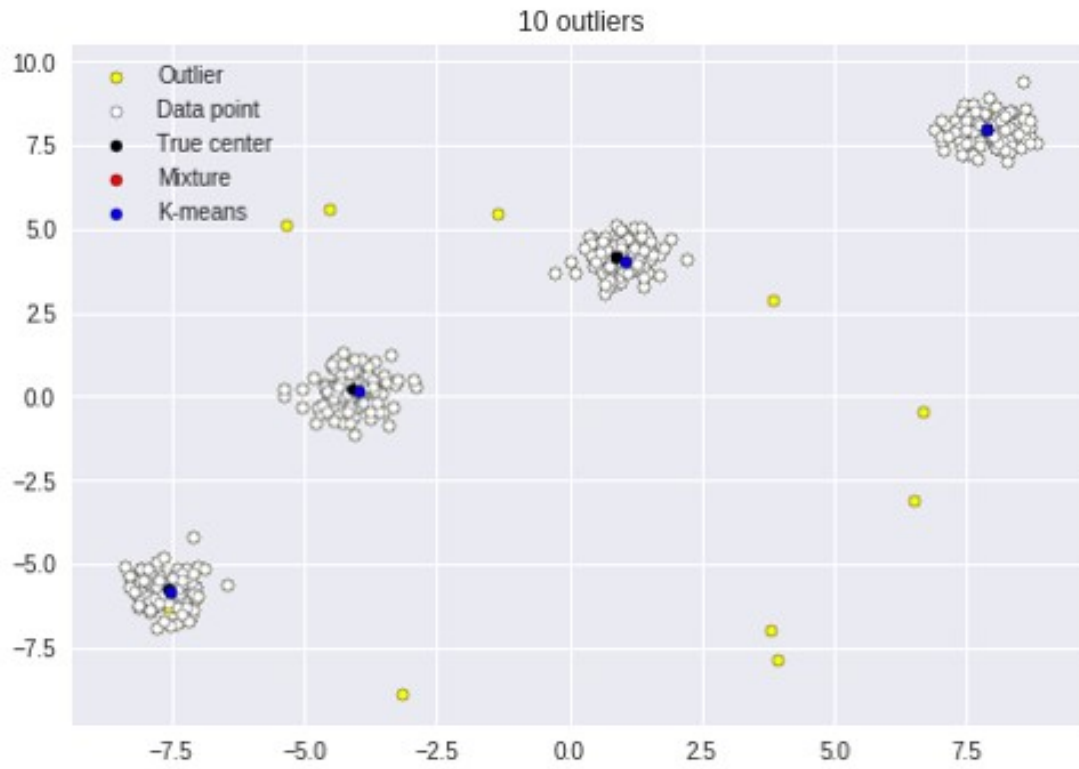


Figure 3.16. Cluster centers and parameter estimates.

Table 3.9. Euclidean distance of cluster centers and parameter estimates with 100 outliers

	Mixture	K-means
Cluster 1	0.59	0.67
Cluster 2	0.02	0.59
Cluster 3	0.00	0.30
Cluster 4	1.26	1.23

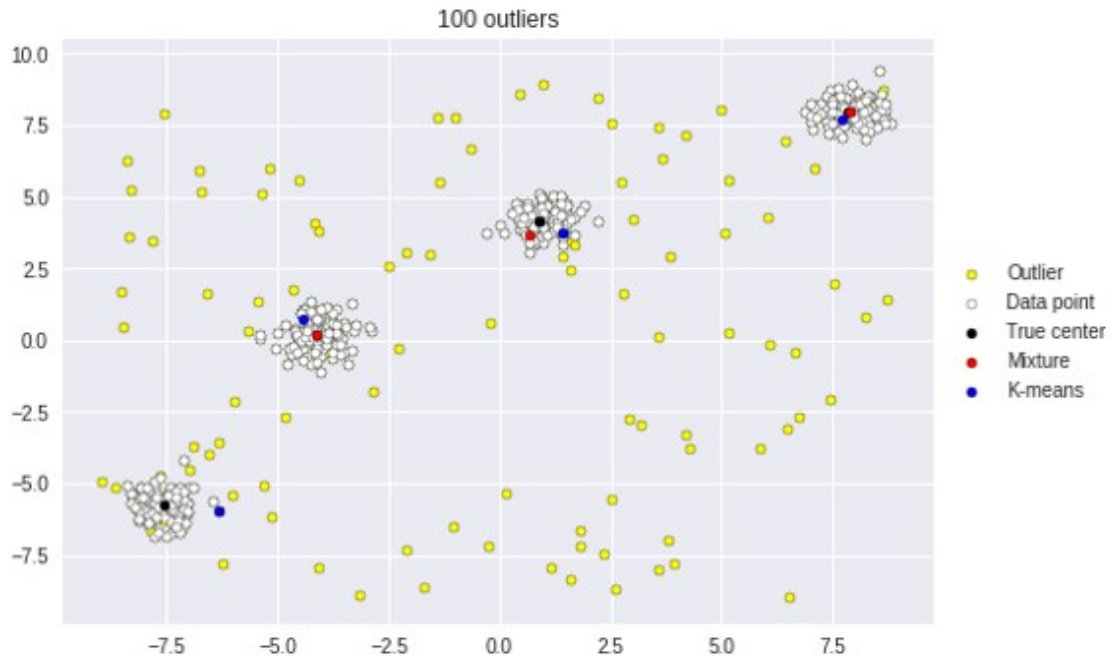


Figure 3.17. Cluster centers and parameter estimates.

Table 3.10. Euclidean distance of cluster centers and parameter estimates with 1000 outliers

	Mixture	K-means
Cluster 1	2.35	3.84
Cluster 2	4.06	2.58
Cluster 3	0.01	4.18
Cluster 4	0.06	1.75

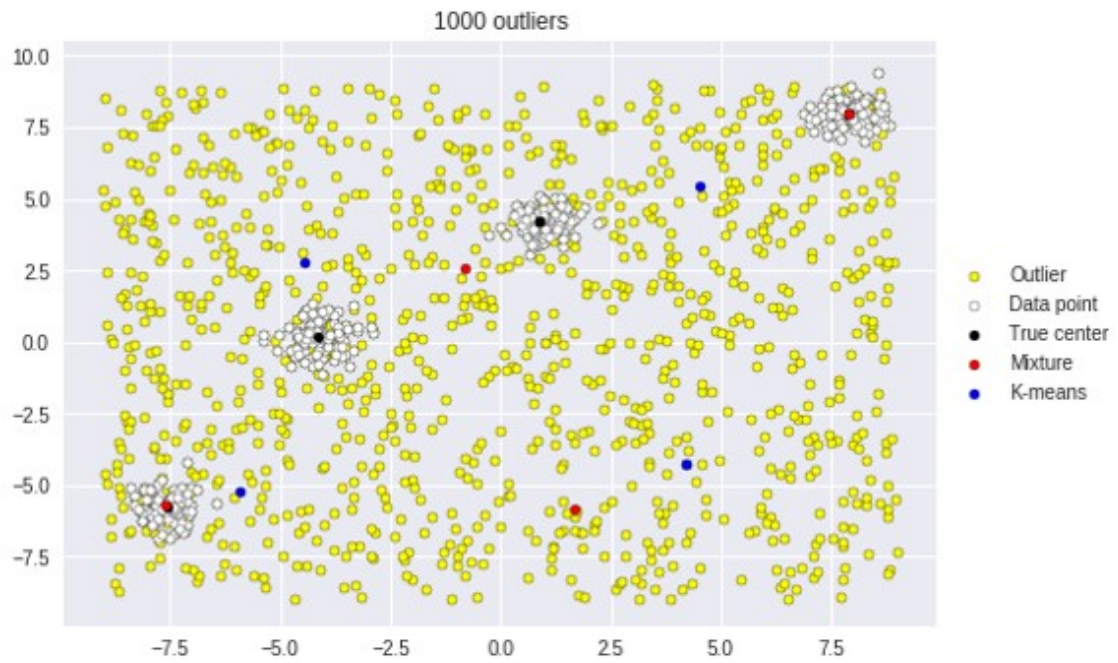


Figure 3.18. Cluster centers and parameter estimates.

Still, as the number of outliers is gets very large, neither of the methods manages to locate cluster centers very well which can be seen in table 3.11 and corresponding figure 3.19.

Table 3.11. Euclidean distance of cluster centers and parameter estimates with 10000 outliers

	Mixture	K-means
Cluster 1	3.51	3.76
Cluster 2	3.66	3.98
Cluster 3	4.66	4.46
Cluster 4	3.27	3.13

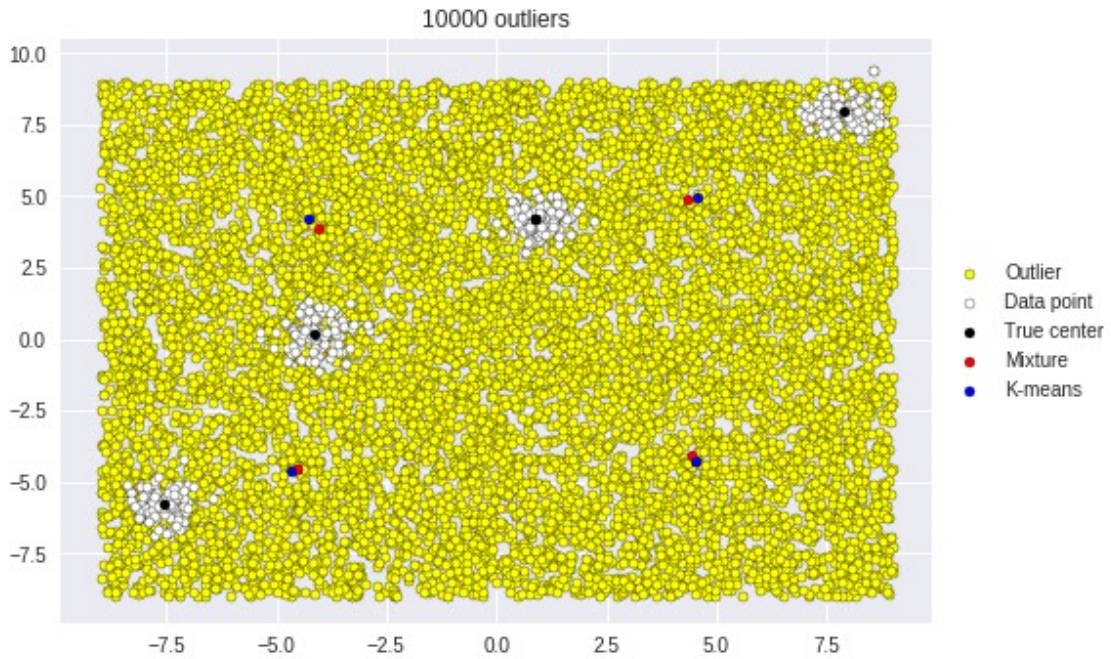


Figure 3.19. Cluster centers and parameter estimates.

Both evaluation approaches (purity and Euclidean distance between cluster centers and estimates) seem to have similar results: both methods seem to do well regardless of the number of observations or the number of dimensions, as purity score was high perfect in these situations and the distance between parameter estimates and true cluster centers was almost non-existent. The differences in performance between the methods appeared in overlapping data and in data with outliers: both methods were affected but mixture models did better in both evaluation approaches.

The previous experiments provided information on changes in individual phenomena separately. As there could be combined effects in some characteristics, an investigation of combined effects of some of these dimensions might provide new information. The combination of, for example, cluster overlap and dimensionality would be an interesting topic to investigate but the combined effect of any of the aforementioned scenarios is not presented in this thesis.

4 Analyzing real data with clustering methods

4.1 Motivation

When the k-means clustering approach was compared to multivariate mixture clustering approach in the previous chapter, the simulations did not account for repeated measurement situations. Albeit it is possible to simulate longitudinal data, actual collected data provide a possibility to use methods in a more realistic situation. In addition to that, the actual number of clusters is not known and thus using real data is a good way to test how both methods choose the optimal number of components. Usage of repeated measurements data also creates an opportunity to use obtained clusters for further analysis. The found clusters can be cross tabulated with a grouping variable that is found in a later time point and that way it is possible to investigate if a cluster membership has relation to participants membership in a group at a later time point. The significance of such a relation can be tested and the p -value can be used as one aspect of comparison between the two methods.

4.2 Data

4.2.1 iLiNS-DYAD

The data used in this thesis was collected for the iLiNS-DYAD (*International Lipid-Based Nutrient Supplements*) project which is a 3 arm trial, in which 1391 pregnant women were enrolled "to assess if healthy growth can be promoted and infant stunting (linear growth retardation) prevented by provision of nutrient and energy-dense lipid-based nutrient supplements (LNS) to the mother-baby dyad during gestation, lactation and in infancy" (Ashorn et al. 2014). In this thesis, the possible effects of the nutrients is not taken into consideration; the focus is only on the growth metrics. For the trial, the wealth of the family was also measured using socio-economic variables as measures for the wealth of the child's family.

During the trial, the children were measured for weight, length, arm circumference, and head circumference. There were seven planned times of measurement: at birth, first week after birth, 26 weeks after birth, 52 weeks after birth, 78 weeks after birth, 104 weeks after birth, and 130 weeks after birth.

4.2.2 Preprocessing the data

There were two variables of interest that were chosen from the whole data for the clustering: weight and length. Both of these variables had seven scheduled measurement times for each enrolled participant (Ashorn et al. 2014). Since there was some variation between the scheduled measurement times and the actual measurement

times that took place, the measurements that were closest to the scheduled date were included. The number of observations included in the analysis is 1245 and there are seven time points.

There were missing values in the data due to participants missing appointments. Missing values were dealt with by imputation and the imputation was carried out using a function from the *mclust* package that applies the EM algorithm for finding the ML estimates for missing values after which the best suited values are augmented to the data. Imputing ML estimates to the data utilizing the EM-algorithm is suitable for multivariate normal data and takes the natural variability in data into account. (Schafer 1997)

The data was treated as multivariate normal with each time point corresponding to a univariate normal vector. In order to use the obtained clusters for prediction, the growth status of the participant at the final time point was determined with z-scores. The participant was classified as "stunted" should the length-for-age z-score (LAZ) be lower than -2 meaning that the child is over two standard deviations shorter than the average. If the participant's LAZ-score was higher than -2 they were classified as "normal". As the growth status was determined by the last measurement, the cluster or trajectory computed from the development of participant's weight can be used to determine if a cluster identity is related to participant's growth status at the last measurement.

4.3 Results

4.3.1 Mixture modeling

Multivariate normal mixture models were first used to cluster the length and weight over six time points for each variable. The ideal number of clusters was determined by using both BIC and ICL measures for model selection. Should the two measures suggest a different number of clusters or different models, a likelihood ratio test would be conducted in order to find the optimal model and optimal number of clusters. In the end, the choice for the number of clusters would be case specific. The modeling was first implemented on the weight variable so that six time points were used for forming the clusters.

The number of clusters suggested by BIC differed from the number of clusters suggested by ICL. According to both BIC and ICL values, the covariance structure would be VEE (Varying variance, ellipsoidal distribution, and equal orientation), but the number of suggested components is different. If model selection was to be made according to BIC value, the number of components would be 3 but with ICL, it would be 2. A likelihood ratio test with bootstrapping was conducted for further insight on model selection. According to the likelihood ratio test with bootstrapping, the optimal number of components would be three. With more evidence suggesting a three-component solution, the model of choice is VEE with three components.

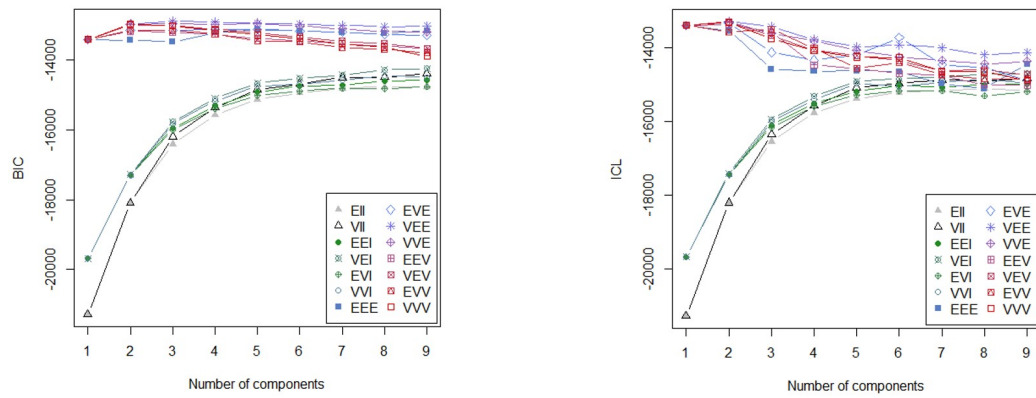


Figure 4.1. BIC and ICL values for each model alternative.

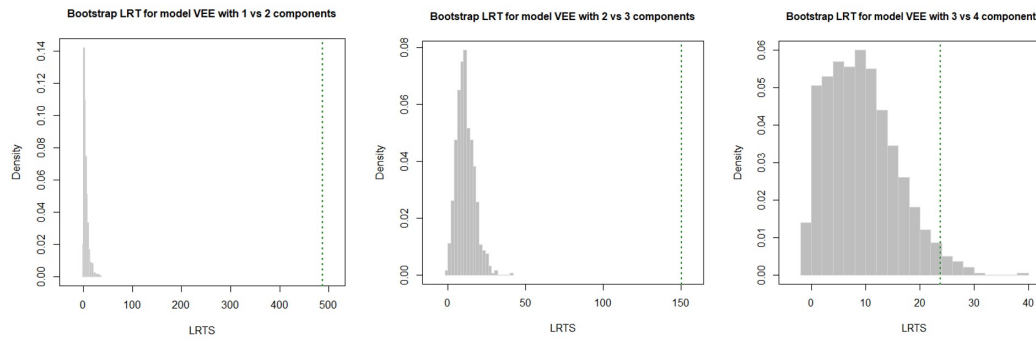


Figure 4.2. Bootstrap likelihood ratio testing. LRTS is the likelihood ratio test statistic and density is the proportion of how many times each value occurred over the bootstrap samples. The vertical line in the pictures is the threshold for significance in the test ie. for significance level of 5%, 95% of the density should be on the left-hand side for the option to be significant.

Table 4.1. Best BIC values

	VEE, 3	VEE, 4	VEE, 5
BIC	-12886.73	-12919.94	-12944.03
BIC diff	0.00	-33.22	-57.30

Table 4.2. Best ICL values

	VEE, 2	VEV, 2	VVE, 2
ICL	-13301.03	-13311.62	-13315.00
ICL diff	0.00	-10.59	-13.97

The clusters can be seen visualized in figure 4.3.

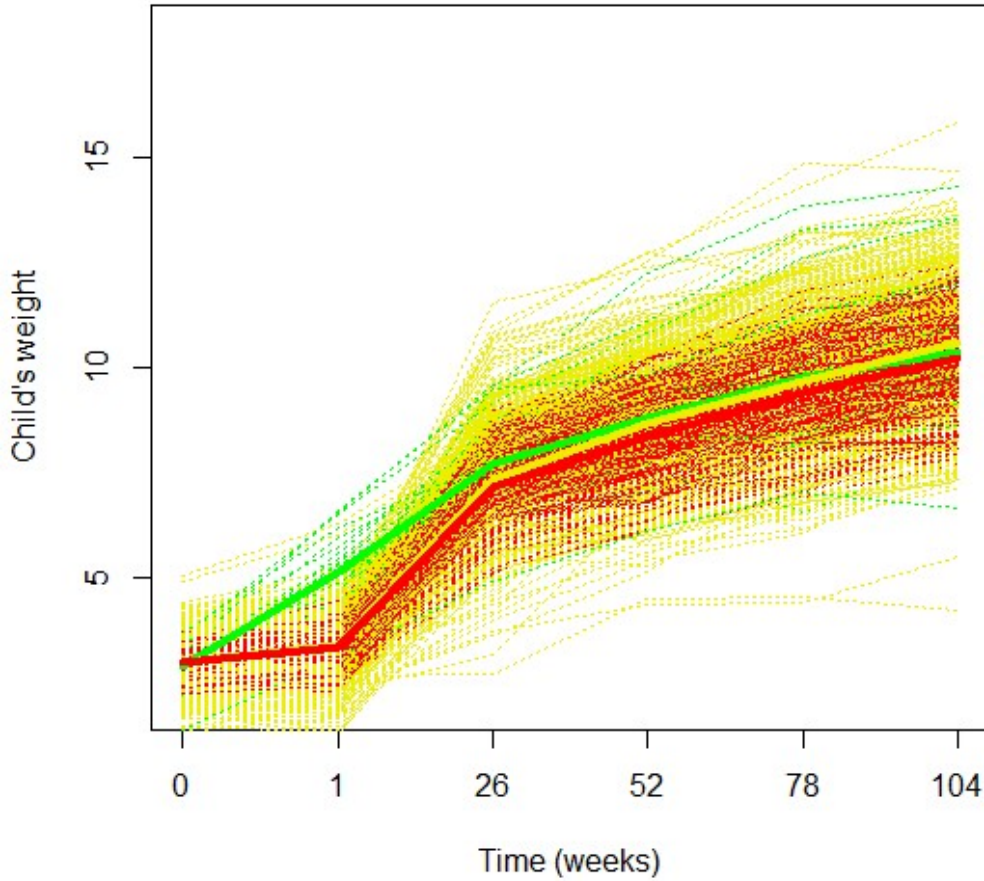


Figure 4.3. Visualization of three clusters

The number of members in each cluster obtained from participant weight was cross tabulated with number of stunted and non-stunted (normal) participants at the seventh time point and a χ^2 test was conducted to test whether the cluster identity is related to the length status at 30 months. The variable from which the outcome classes were formed was not imputed so the prediction is done only on the data that was available. The χ^2 test showed no significance with cluster affiliation and stuntedness, even though the p -value was quite small ($p = 0.052$). This would suggest that a membership in a cluster calculated from the weight of the participant through six time points from birth to 104 weeks doesn't have a relationship with whether or not the participant is stunted at the 30 months time point.

However, as clustering was then conducted five times again, each time using one time point less than in previous clustering, significance was discovered. As seen in tables 4.4 and 4.5, the cluster identity is significantly related to the growth status at

the final time point. The significance is not found if less than four time points are used for forming the clusters (see tables 4.6 to 4.8). The model choice for all the time points was obtained using BIC as the model selection criterion.

Table 4.3. *Mixture models:* Table of clusters attained using first six time points

	Normal	Stunted
Cluster 1 (n = 14)	5	9
Cluster 2 (n = 242)	140	102
Cluster 3 (n = 340)	167	173
$\chi^2 = 5.913$, df = 2, p-value = 0.052		

Table 4.4. *Mixture models:* Table of clusters attained using first five time points

	Normal	Stunted
Cluster 1 (n = 190)	112	78
Cluster 2 (n = 393)	196	197
Cluster 3 (n = 13)	4	9
$\chi^2 = 6.709$, df = 2, p-value = 0.035		

Table 4.5. *Mixture models:* Table of clusters attained using first four time points

	Normal	Stunted
Cluster 1 (n = 414)	202	212
Cluster 2 (n = 166)	102	64
Cluster 3 (n = 11)	3	8
$\chi^2 = 9.737$, df = 2, p-value = 0.008		

Table 4.6. *Mixture models:* Table of clusters attained using first three

	Normal	Stunted
Cluster 1 (n=124)	77	47
Cluster 2 (n=444)	221	223
Cluster 3 (n=28)	14	14
$\chi^2 = 5.965$, df = 2, p-value = 0.051		

Table 4.7. *Mixture models:* Table of clusters attained using first two time points

	Normal	Stunted
Cluster 1 (n=464)	237	227
Cluster 2 (n=24)	13	11
Cluster 3 (n=108)	62	46
$\chi^2 = 1.440$, df = 2, p-value = 0.487		

Table 4.8. *Mixture models:* Table of clusters attained using the first time point

	Normal	Stunted
Cluster 1 (n=127)	70	57
Cluster 2 (n=469)	242	227
$\chi^2 = 0.365$, df = 1, p-value = 0.546		

Even if a relation to length-wise growth status couldn't have been discovered, the clusters in themselves give information on the weight development of the child, which can be vital in detecting associations affecting the weight of the child. From graph 4.3 one can detect three lines, in green, yellow, and red that represent the mean of each cluster formed from the weight variable. It would seem that there are three groups in the data: the "high-weight" children, the "mid-weight" children and the "low-weight" children.

A powerful tool for inference for the parameter estimates, bootstrapping, can be utilized to calculate the 95% confidence interval for the mean weight of the children at each time point. Confidence intervals for each cluster can be found from tables 4.9 to 4.11

Table 4.9. Bootstrap confidence intervals of "high-weight" cluster

Obs: 30	Time 1	Time 2	Time 3	Time 4	Time 5	Time 6
2.5%	2.69	4.50	7.17	8.32	9.23	9.86
97.5%	3.08	5.68	8.13	9.35	10.58	11.19

Table 4.10. Bootstrap confidence intervals of "Mid-weight" cluster

Obs: 548	Time 1	Time 2	Time 3	Time 4	Time 5	Time 6
2.5%	2.87	3.26	7.17	8.47	9.49	10.36
97.5%	2.99	3.43	7.42	8.73	9.76	10.65

Table 4.11. Bootstrap confidence intervals of "Low-weight" cluster

Obs: 667	Time 1	Time 2	Time 3	Time 4	Time 5	Time 6
2.5%	2.94	3.27	7.07	8.30	9.33	10.17
97.5%	3.01	3.35	7.25	8.49	9.53	10.39

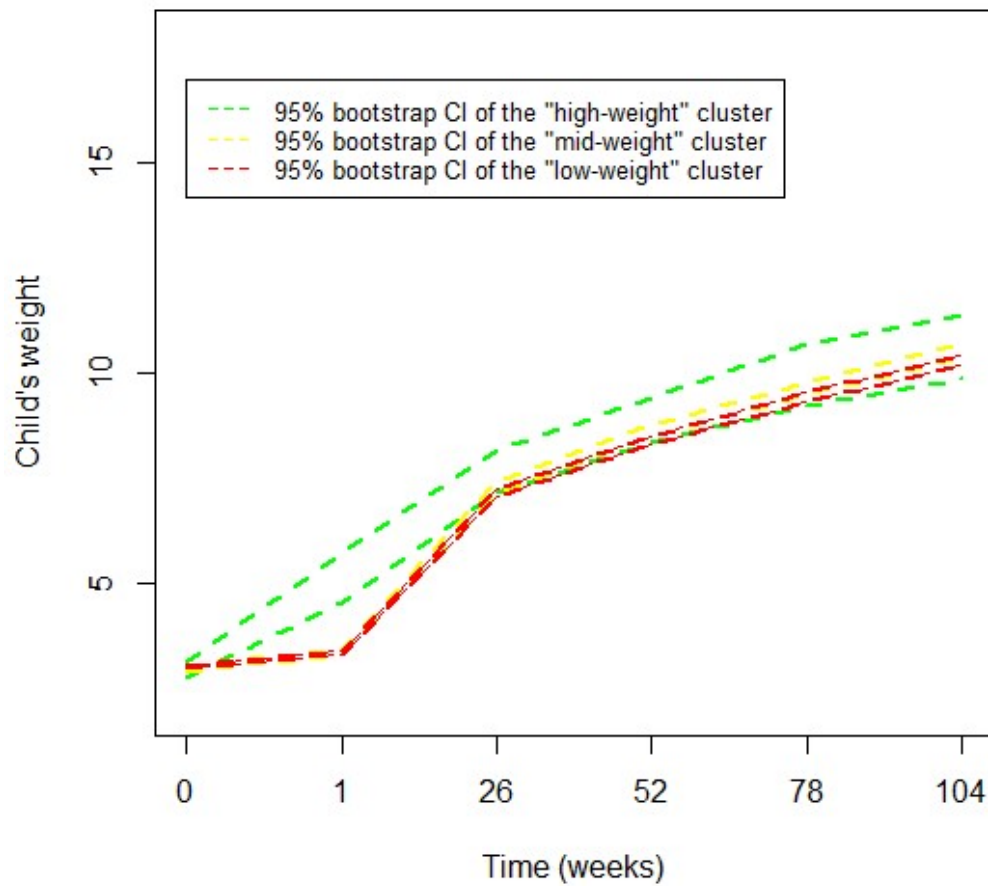


Figure 4.4. Visualization of 95% bootstrap confidence intervals for parameter estimates

4.3.2 KmL

K-means clustering for longitudinal data was applied on the data for comparison purposes as well as to validate the previously obtained clusters and relation of cluster membership and growth status at later data points. The stages that were done with mixture models were repeated with KmL as well.

The *KmL* algorithm automatically tests different numbers of components and different initialization methods before suggesting an optimal number of clusters (Genolini et al. 2010). The k-means approach suggested two clusters as the optimal number of clusters using the Calinski-Harabasz Criterion when six time points were used.

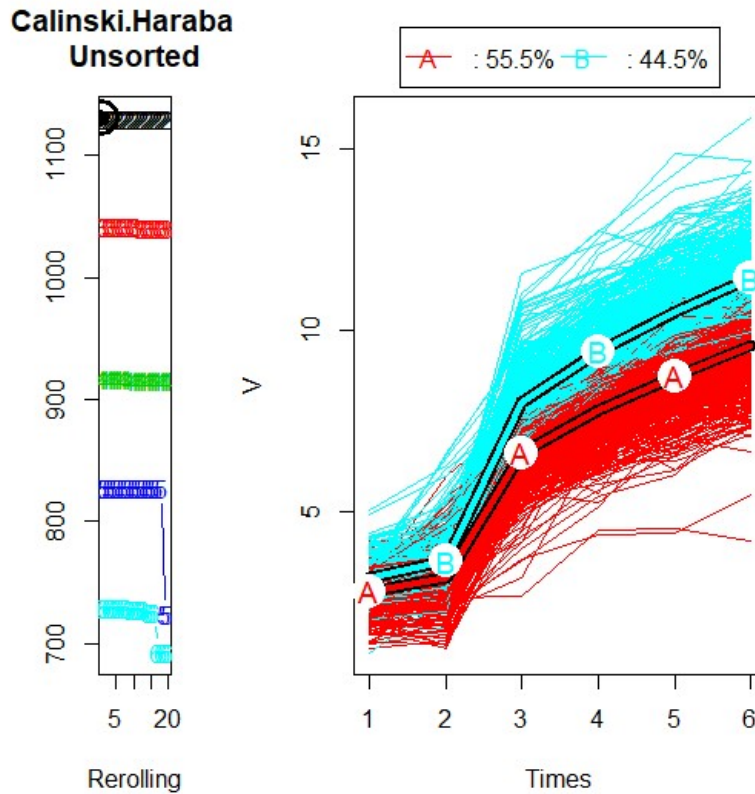


Figure 4.5. Clusters obtained with *KmL* and Calinski-Harabasz score

The χ^2 test yields a similar result as with mixture models, suggesting that the cluster identity of weight development of a participant is related to the stuntedness of a participant at 30 months. However, unlike with mixture models the relation between cluster identity and stuntedness at 30 months was significant regardless of how many time points were used for forming the clusters.

Table 4.12. *K-means*: Table of clusters attained using first six time points

	Normal	Stunted
Cluster 1 (n=326)	111	215
Cluster 2 (n=270)	201	69
$\chi^2 = 94.996$, df = 1, p-value < 0.001		

Table 4.13. *K-means*: Table of clusters attained using first five time points

	Normal	Stunted
Cluster 1 (n=313)	108	205
Cluster 2 (n=284)	77	92
$\chi^2 = 82.643$, df = 1, p-value < 0.001		

Table 4.14. *K-means*: Table of clusters attained using first four time points

	Normal	Stunted
Cluster 1 (n=310)	109	201
Cluster 2 (n=286)	203	83
$\chi^2 = 103.87$, df = 5, p-value < 0.001		

Table 4.15. *K-means*: Table of clusters attained using first three time points

	Normal	Stunted
Cluster 1 (n=328)	128	200
Cluster 2 (n=268)	184	84
$\chi^2 = 50.736$, df = 1, p-value < 0.001		

Table 4.16. *K-means*: Table of clusters attained using first two time points

	Normal	Stunted
Cluster 1 (n=313)	137	176
Cluster 2 (n=283)	175	108
$\chi^2 = 29.685$, df = 5, p-value < 0.001		

Table 4.17. *K-means*: Table of clusters attained using the first time point

	Normal	Stunted
Cluster 1 (n=306)	180	126
Cluster 2 (n=290)	132	158
$\chi^2 = 10.042$, df = 1, p-value = 0.002		

5 Conclusion

In this thesis multivariate mixture models were explored and compared to a k-means based clustering method in both simulated and real data. There were differences in the performance of the methods in simulations as well as in real data.

When experimented on simulated data, neither of the methods were affected by increased number of observations when the clusters were well separated. Increase in overlap decreased the purity score of both methods but with full covariance structure specified for mixture models, mixture models were able to perform better with increasing cluster overlap. Increase in dimensionality didn't affect either of the methods when clusters were well separated. As the number of outliers was increased, mixture models performed clearly better than k-means. These scenarios gave some evidence on mixture models performing better in scenarios where both methods had deteriorating performance.

Mixture models and k-means clustering are both powerful tools for clustering and can even be utilized for longitudinal data. When applied on real data that was collected from Malawian children over a 30 month period, the methods gave different suggestions for the optimal number of components. By applying k-means clustering for longitudinal data using six time points, the two cluster option determined by Calinski-Harabasz criterion was deemed the best option whereas for multivariate normal mixture models, the optimal number of components was three according to BIC. With both methods it seemed that the more time points were used for the cluster model the stronger the significance was between the cluster identity in weight and growth status at 30 months. *KmL*, however, formed clusters that had statistical significance to relation to growth status using any number of time points whereas multivariate normal mixture models required at least four time points to form clusters in which the cluster identity had statistical significant relation to growth status at the last time point.

An important question to ask oneself when comparing the two methods is what is the method being used for. K-means would seem to be a good option for dividing data into well separated groups. In the situation of k-means, the formed clusters will be clearly separated which can be a valued attribute in results in some cases. However, this is a problem if there is a possibility for overlapping of the groups in the data and one is looking for the "ground truth" clusters. If the repeated measurement data used in this thesis is considered, the k-means based method divided the participants into clearly divided clusters which is a useful result if the aim would be divide participants into two groups: participants that have positive development of weight and those who do not. With mixture models on the other hand, the optimal component solution was deemed to be one with three clusters and the clusters were not as clearly separated based on the growth paths of the participants. This is not necessarily a weakness or an error as it is possible that mixture models were more successful in identifying developmental paths that can be, in fact, overlapping.

As *KmL* was able to form two clusters for the weight development, it is natural

that those two groups could predict stuntedness of a participant when there are only two groups in the prediction variable. Mixture models were also able to obtain significant results in terms of cluster-identity having relation to growth status at the last time point and it is also possible that if there were three classes in the outcome variable, mixture models could have been able to identify that middle class. Still, if that were the situation *KmL* could have also have found three clusters that could have had significant relation to the outcome growth status, even though this wouldn't have been the optimal number of components when using Calinski-Harabasz criterion.

Combining the information obtained from the experiments with simulations to the analysis on real world data it could be said that mixture models are better suited for describing the "true" developmental paths of an individual. The simulation experiments suggested that mixture models estimate the cluster centers better when there is overlapping or outliers in the data which is not an uncommon situation in actual, real world data. Thus, one could argue that in the real world situation described in this thesis, the optimal number of clusters obtained with k-means based methods doesn't detect the subtle nuances that mixture models can take into account by modeling the covariance structure. That being said, *KmL* is a very sophisticated method developed precisely for longitudinal data and it has been developed with usual clustering issues in mind. Simulations in a longitudinal context would be useful in future comparisons between the methods. This thesis also didn't test a modified Cholesky-decomposed version in practice and that would possibly be a useful addition to future research.

Both mixture models and *KmL* were able to find clusters that seemed to predict dependence of cluster-identity and growth status. Nonetheless, if prediction was the task for which the mixture models and *KmL* were used, further exploration of different models and numbers of components would have been beneficial. In this thesis, the prediction was an added dimension to the comparison between the methods. The prediction capacity of both methods is something that could also be an interesting topic for further investigation.

Furthermore, for utilizing the found clusters of unclassified data, one could use multinomial logistic regression for identifying variables that significantly affect the data point belonging to a cluster or use the clusters as predictors in linear models for example. There are many possibilities for using the found clusters and one could benefit from using more than one method for validating the obtained results. Nevertheless, based on the experiments conducted in this thesis one could argue that it would be beneficial to use mixture models for clustering, especially if one expects the data to have outliers or the clusters to overlap. One should still bear in mind that these experiments were carried out using Gaussian data. Comparison of the methods in a non-Gaussian setting would also be an interesting study topic.

References

- Ashorn P., Alho L., Ashorn U., Cheung Y.B., Dewey K.G., Harjunmaa U., Lartey A., Nkhoma M., Phiri N., Phuka J., Vosti S.A., Zeilani M., Maleta K. (2014), "The impact of lipid-based nutrient supplement provision to pregnant women on newborn size in rural Malawi: A randomised controlled trial", *Am J Clin Nutr.* Feb;101(2):387-97. doi: 10.3945/ajcn.114.088617. Epub 2014 Dec 10.
- Baid U., Talbar S. (2017), "Comparative Study of K-means, Gaussian Mixture Model, Fuzzy C-means algorithms for Brain Tumor Segmentation", *Advances in Intelligent Systems Research.* Vol. 137, pp. 592-597. Atlantis Press
- Banfield J., Raftery A. (1993), "Model-Based Gaussian and Non-Gaussian Clustering" *Biometrics*, vol. 49, no. 3, 1993, pp. 803 - 821. JSTOR, www.jstor.org/stable/2532201.
- Biernacki C., Celeux G., and Govaert G. (2000), "Assessing a mixture model for clustering with the integrated completed likelihood", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(7):719-725.
- Bock H. (2000), "Origins and extensions of the k-means algorithm in cluster analysis", *Journal Electronique d'Histoire des Probabilités et de la Statistique*, Vol 4, n. 2, 2008 Available from the Internet: <http://www.jehps.net/Decembre2008/Bock.pdf>
- Browne R., & McNicholas P. (2014), "Estimating common principal components in high dimensions", *Advances in Data Analysis and Classification*, 8(2):217 - 226, 2014
- Calinski T., Harabasz J. (1974), "A dendrite method for cluster analysis", *Commun Stat* 3(1):1-27
- Celeux G., & Govaert G., (1995), "Gaussian parsimonious clustering models" *Pattern Recognit.* 28.
- Cramèr H. (1946), "Mathematical Methods of Statistics", Princeton N.J.: Princeton University Press.
- De la Cruz-Mesia R., Quintana F. A. & Marshall G., "Model-based clustering for longitudinal data", *Computational Statistics & Data Analysis*, v.52 n.3, pp.1441–1457 [doi:10.1016/j.csda.2007.04.005] Available from the Internet: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.332.258&rep=rep1&type=pdf>
- Dempster, A.P., Laird, N.M., Rubin, D.B. (1977), "Maximum Likelihood from Incomplete Data via the EM Algorithm", *Journal of the Royal Statistical Society, Series B.* 39 (1): 1-38. JSTOR 2984875. MR 0501537.
- Fraley C., & Raftery A. (2002), "Model-Based Clustering, Discriminant Analysis and Density Estimation", *Journal of the American Statistical Association*, 97:611 - 631.
- Greene, W.H. (1990), "Econometric Analysis", New York: Macmillan.
- Genolini C., Alacoque X., Sentenac M., Arnaud C. (2015), "kml and kml3d: R Packages to Cluster Longitudinal Data", *Journal of Statistical Software*, 65(4)
- Genolini, C. & Falissard B. (2010), 'KmL: k-means for longitudinal data', *Computational Statistics*, 2:317–328. Available from the Internet: [http://christophe.genolini.free.fr/recherche/aTelecharger/Genolini%20\(2010\)%20KmL%20K-means%20for%20Longitudinal%20Data.pdf](http://christophe.genolini.free.fr/recherche/aTelecharger/Genolini%20(2010)%20KmL%20K-means%20for%20Longitudinal%20Data.pdf)

- Goldfeld, K. (2018), "simstudy: Simulation of Study Data", R package version 0.1.9. Available from the Internet: <https://CRAN.R-project.org/package=simstudy>
- Hartigan, J. A. & Wong, M. A. (1979), "Algorithm AS 136: A K-Means Clustering Algorithm", *Journal of the Royal Statistical Society, Series C*.
- Jones, B., Nagin D., and Roeder K. (2001) "A SAS Procedure Based on Mixture Models for Estimating Developmental Trajectories." *Sociological Research and Methods*, 29: 374-393.
- Kim, H. & Park H. (2007), "Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis." *Bioinformatics* (Oxford, England), *23*(12), pp. 1495-502. ISSN 1460-2059
- Leisch, F. (2003), "FlexMix: A general framework for finite mixture models and latent class regression in R", Available from the Internet: <https://cran.r-project.org/web/packages/flexmix/vignettes/flexmix-intro.pdf>
- MacQueen, J. B. (1967). *Some Methods for classification and Analysis of Multivariate Observations*. Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability. 1. University of California Press.
- Maitra, R. & Melnykov, V. (2010), "Simulating data to study performance of finite mixture modeling and clustering algorithms", *The Journal of Computational and Graphical Statistics*, 19(2):354-376.
- McLachlan, G. (1987), "On bootstrapping the likelihood ratio test statistic for the number of components in a normal mixture", *Applied statistics*, 36:318 - 324
- McLachlan, G. & Peel, D. (2000), "Finite Mixture Models", Wiley-Interscience Publication.
- McNicholas, P.D. & Murphy, T. B. (2010), "Model-based clustering of longitudinal data", *The Canadian Journal of Statistics* 38(1), pp. 153–168. Available from the Internet: <http://irserver.ucd.ie/bitstream/handle/10197/2834/cdgm22.pdf?sequence=1>
- Nagin, D.S. (2005), "Group-Based Modeling of Development", Harvard University Press.
- Nagin, D.S. & Odgers, C.L. (2010), "Group-Based Trajectory Modeling in Clinical Research", *Annual Review of Clinical Psychology*, 6, 109-138. Available from Internet: https://ssrc.indiana.edu/doc/wimdocs/2013-03-29_nagin_nagin_odgers_article.pdf.
- MacKay, D.(2003), "Information Theory, Inference and Learning Algorithms" , Cambridge University Press.
- Pedregosa et al.(2011), "Scikit-learn: Machine Learning in Python", *JMLR* 12, pp. 2825-2830, 2011
- Qiu D. (2010), "A comparative study of the K-means algorithm and the normal mixture model for clustering: Bivariate homoscedastic case", *Journal of Statistical Planning and Inference*, 140(7):1701-1711
- Reinecke J. & Seddig D. (2011), "Growth mixture models in longitudinal research", *Advances in Statistical Analysis* 95:415–434, doi: 10.1007/s10182-011-0171-4
- Schafer J., (1997) "Analysis of Incomplete Multivariate Data", Chapman & Hall
- Scrucca L., Fop M., Murphy T. B. & Raftery A. E. (2017), "mclust 5: clustering, classification and density estimation using Gaussian finite mixture models", *The R Journal* 8(1):205-233 Available from Internet: <https://journal.r-project.org/archive/2016/RJ-2016-021/RJ-2016-021.pdf>
- Steinley, D., & Brusco, M. (2011), "Evaluating Mixture Modeling for Clustering: Recommendations and cautions." *Psychological Methods*, 16:63-79

- VanderPlas J.(2016), "Python Data Science Handbook", O'Reilly Media
- Vermunt J. K. (2010), "Longitudinal Research Using Mixture Models" in *Longitudinal Research with Latent Variables*, eds. K. van Monfort, J. H. Oud & A. Satorra, Springer-Verlag Berlin Heidelberg, pp. 119–152
- Vermunt J. K. (2011), "K-Means May Perform as Well as Mixture Model Clustering but May Also Be Much Worse : Comment on Steinley and Brusco (2011)", *Psychological Methods*, 16 (1):82-88
- Thiel H. (1971), "Principles of Econometrics", New York: Wiley.

Appendix A: Histograms for normality

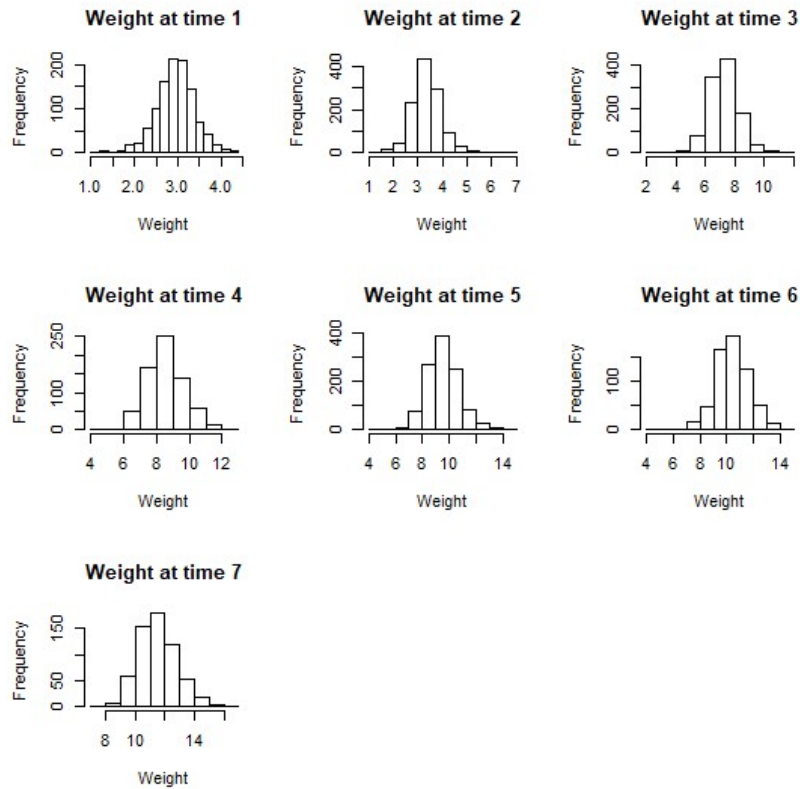


Figure 1. Histograms of weight at each time point